Monadic Regions: Formal Type Soundness and Correctness *

Matthew Fluet
Dept. of Computer Science
Cornell University
Ithaca, NY 14853
fluet@cs.cornell.edu

April 22, 2004

Abstract

Drawing together two lines of research (that done in type-safe region-based memory management and that done in monadic encapsuation of effects), we give a type-preserving translation from a variation of the region calculus of Tofte and Talpin into an extension of System F augmented with monadic types and operations. Our source language is a novel region calculus, dubbed the Single Effect Calculus, in which sets of effects are specified by a single region representing an upper bound on the set. Our target language is $\mathsf{F}^{\mathsf{RGN}}$, which provides an encapsulation operator whose parametric type ensures that regions (and values allocated therein) are neither accessible nor visible outside the appropriate scope.

1 Introduction

Region-based memory management is a particular way to manage the dynamically (or *heap*) allocated memory of a program. It stands in contrast to explicit memory management by the programmer using operations like C's `malloc` and `free` and to fully automatic memory management using a garbage collector. In a region-based memory management system, regions are areas of memory holding heap allocated data. Regions have syntactic lifetimes, following the block structure of the program. A region is created upon entering a region delimited block; for the duration of the block, data can be allocated into the region; upon exiting the block, the entire region (including all data allocated within it) is destroyed. Tofte and Talpin's *region calculus* [25, 26] introduced a type system, based on effects, that ensures the safety of this allocation and deallocation mechanism. A unique feature of this scheme is that evaluation can lead to *dangling pointers*: a pointer to data that has been deallocated. So long as the program never dereferences such a pointer (a fact that the type system verifies), the program can be safely run. This aspect of region-based memory management systems can lead to better memory usage than that achieved by garbage collectors, which do not allow dangling pointers, on some programs.

While there has been some work aimed at integrating garbage collection and region-based memory management [9, 6], it is not possible to manage particular data by one scheme or the other. Cyclone [12] offers multiple forms of memory management in the context of a safe dialect of C. However, it makes use of a sophisticated type-and-effect system to ensure safety. We seek, therefore, a simple account that supports region-based memory management within a "traditional" functional programming language that primarily relies upon garbage collection and a simple type system.

A separate line of research has investigated mechanisms by which imperative (and otherwise "foreign") constructs can be safely integrated into pure functional languages. Launchbury and Peyton Jones [14, 15] introduced monadic state as a means by which imperative computations could be embedded in the pure

*A preliminary version of this work appeared in the informal *Proceedings of the 2nd Workshop on Semantics, Program Analysis, and Computing Environments for Memory Management (SPACE04).*

evaluation of a functional program. The encapsulation operator runST has a type that statically guarantees that stateful computations appear as pure functions to the rest of the program. Inspired by this work, we propose monadic regions as a mechanism for embedding region-based memory management within a pure functional language.

The main contributions of this paper are three-fold. First, we introduce a novel region calculus, dubbed the *Single Effect Calculus*. The Single Effect Calculus tracks a partial order on regions; this order can be used allow a single region to bound the effect of a set of regions. Second, we introduce a novel monadic language, dubbed $\mathsf{F}^{\mathsf{RGN}}$, which is an extension of System F that adds monadic types and operations for manipulating regions. A key aspect of this language is that no extension (beyond adding three type constructors) to the type system of System F is required; in particular, type equality in $\mathsf{F}^{\mathsf{RGN}}$ is syntactic and all new language expressions can be interpreted as constants with polymorphic types. Encapsulation of region computations in $\mathsf{F}^{\mathsf{RGN}}$ is ensured by the type system, using parametricity. Finally, we give a type preserving translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$, an important first step towards demonstrating the adequacy of $\mathsf{F}^{\mathsf{RGN}}$ for expressing region-based memory management.

The remainder of this paper is structured as follows. In the next two sections, we describe the Single Effect Calculus and $\mathsf{F}^{\mathsf{RGN}}$. Section 4 presents a type preserving translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$. Section 5 explores the expressiveness of the Single Effect Calculus. In Section 6, we consider related work. Section 7 concludes and notes some directions for future work.

	i	$\in$	$\mathbb{Z}$
	ϑ, ϱ	$\in$	$RVars^{SEC}$ where $\mathcal{H} \in RVars^{SEC}$
	f, x	$\in$	$Vars^{SEC}$
Surface region placeholders	θ, ρ	$::=$	ϱ
Surface effects	φ	$::=$	$\{\rho_1, \ldots, \rho_n\}$
Surface types	τ	$::=$	$\mathsf{bool} \mid (\mu, \rho)$
Surface boxed types	μ	$::=$	$\mathsf{int} \mid \tau_1 \xrightarrow{\theta} \tau_2 \mid \tau_1 \times \tau_2 \mid \Pi \varrho \succeq \varphi.^{\theta} \tau$
Surface programs	p	$::=$	e
Surface terms	e	$::=$	i at $\rho \mid e_1 \oplus e_2$ at $\rho \mid e_1 \ominus e_2 \mid$ tt $\mid$ ff $\mid$ if e_b then e_t else $e_f \mid$ $x \mid \lambda x : \tau.^{\theta} e$ at $\rho \mid e_1\ e_2 \mid$ (e_1, e_2) at $\rho \mid$ fst $e \mid$ snd $e \mid$ letregion ϱ in $e \mid \lambda \varrho \succeq \varphi.^{\theta} u$ at $\rho \mid e\ [\rho]$
Surface abstractions	u	$::=$	$\lambda x : \tau.^{\theta} e$ at $\rho \mid \lambda \varrho \succeq \varphi.^{\theta} u$ at ρ
Surface values	v	$::=$	tt $\mid$ ff $\mid x$

Figure 1: Surface syntax of SEC

2 Single Effect Calculus

The Single Effect Calculus is a variation of the region calculus of Tofte and Talpin [25, 26], in the spirit of [11], and taking inspiration from the Capability Calculus [5] and Cyclone [8]. Essentially, the Single Effect Calculus capitalizes on the fact that a LIFO stack of regions imposes a partial order on live (allocated) regions. Regions lower on the stack outlive regions higher on the stack. Hence, the liveness of a region implies the liveness of all regions below it on the stack. Thus, it is the case that a single region can serve as a witness for a set of effects: the region appears as a *single effect* in place of the set.

We present a full formalism of the Single Effect Calculus and a syntactic proof of type preservation. We begin with a presentation of the language (surface syntax, computation syntax, dynamic semantics, and static semantics) and then proceed to the proof.

The dynamic semantics defines a large-step (or natural) semantics, which defines an *evaluation relation* from *stacks of regions* and *expressions* to *stacks of regions* and *values*.

The proof of type preservation is presented in "bottom-up" order, where all relevant lemmas are presented (and proved) before being used.

2.1 Surface Syntax of SEC

Figure 1 presents the syntax of "surface programs" (that is, excluding intermediate terms that will appear in the operational semantics) of the Single Effect Calculus. In the following sections, we explain and motivate the main constructs and typing rules of the Single Effect Calculus.

2.2 Types

In Tofte and Talpin's original region calculus, a region is associated with every type that requires heap allocated storage. We assume that integers, pairs, and function closures require heap allocated storage, while booleans do not. The type (μ, ρ) pairs together a boxed type (a type requiring heap allocated storage) and a place (a region); we interpret (μ, ρ) as the type of objects of boxed type μ allocated in region ρ. For our external language, it suffices to allows places to range over region variables ($RVars^{SEC}$), which include a distinguished member $\mathcal{H}$, corresponding to a global region that remains live (allocated) throughout the

execution of the program. Various operational semantics extend the syntactic class of places to include concrete region names [26] and/or a special constant corresponding to a deallocated region [4]; we will do so in the next section.

Of the boxed types, integers and pairs are standard. More interesting are the two boxed types corresponding to abstractions. Recall that in previous formulations of region calculi, the types of functions and region abstractions are given by:

$$\tau_1 \xrightarrow{\varphi} \tau_2 \quad \text{and} \quad \Pi\varrho.^{\varphi}\tau$$

where φ is an effect, a finite set of places. In the function and region-abstraction types, the effect φ denotes a *latent effect*, the set of regions read from or written to when an argument is supplied to the function or a region is supplied to the region abstraction.

Our formulation of function and region abstraction types differs. The type of a function is given by:

$$\tau_1 \xrightarrow{\theta} \tau_2$$

where θ is a (single) place. Whereas φ denoted the set of regions affected by executing the function, θ denotes an upper bound (in the partial order of regions) on the set of regions affected when the function is applied to an argument. In the case of the former, the typing judgement for an application requires showing that all regions in φ are live. In the case of the latter, the typing judgement for an application only requires showing that θ is live; the liveness of all regions below θ are implied by the stack discipline.

The type of a region abstraction is given by:

$$\Pi\varrho \succeq \varphi.^{\theta}\tau$$

where φ and θ are a finite set of places and a place, respectively. (Note that the region variable ϱ is bound within θ and τ, but not φ.) The bounded quantification in this type requires that at the instantiation of ϱ by a region ρ, we must be able to show that the livness of ρ implies the liveness of all the regions in φ. Within the body of the abstraction, we assume that ϱ is an upper bound on the set of regions φ. As with a function type, θ denotes an upper bound on the set of regions affected when the region abstraction is applied to a region.

2.3 Programs and Terms

Programs in the Single Effect Calculus are simply terms. We distinguish programs as a syntactic class because the type system presented in the next section has a special judgement for top-level programs. Essentially, this judgement establishes reasonable "boundary conditions" for a program's execution, an aspect that is often overlooked in other descriptions of region calculi.

Terms in the Single Effect Calculus are similar to those found in the λ-calculus. One major difference is that terms yielding heap allocated values carry a region annotation $\mathsf{at}\ \rho$, which indicates the region in which the value is to be allocated. New regions are introduced (and implicitly created and destroyed) by the $\mathsf{letregion}\ \varrho\ \mathsf{in}\ e$ term. The region variable ϱ is bound within e, demarcating the scope of the region. Within e, values may be read from or allocated in the region ϱ. Executing $\mathsf{letregion}\ \varrho\ \mathsf{in}\ e$ allocates a new region of memory, then executes e, and finally deallocates the region.

The term $\lambda\varrho \succeq \varphi.^{\theta}u\ \mathsf{at}\ \rho$ introduces a region abstraction (allocated in the region ρ), where the term u is polymorphic in the region ϱ.[1] Such region polymorphism is particularly useful in the definition of functions, in which we parameterize over the regions necessary for the evaluation of the function. As explained in the previous section, region abstractions make use of bounded quantification; the intention is that ϱ is an upper bound on th set of regions regions φ. The term $e\ [\rho]$ eliminates a region abstraction; operationally, it substitutes the place ρ for the region variable ϱ in u and evaluates resulting term.

	i	$\in$	$\mathbb{Z}$	
	ϑ, ϱ	$\in$	$RVars^{SEC}$	where $\mathcal{H} \in RVars^{SEC}$
	f, x	$\in$	$Vars^{SEC}$	
	l	$\in$	$Locs^{SEC}$	
Region names	r	$\in$	$RNames$	where $\mathsf{H} \in RNames^{SEC}$
Computation region placeholders	θ, ρ	$::=$	$\varrho \mid r \mid \bullet$	
Computation effects	φ	$::=$	$\{\rho_1, \ldots, \rho_n\}$	
Computation types	τ	$::=$	$\mathsf{bool} \mid (\mu, \rho)$	
Computation boxed types	μ	$::=$	$\mathsf{int} \mid \tau_1 \xrightarrow{\theta} \tau_2 \mid \tau_1 \times \tau_2 \mid \Pi \varrho \succeq \varphi.^{\theta} \tau$	
Computation programs	p	$::=$	e	
Computation terms	e	$::=$	$i \ \mathsf{at}\ \rho \mid e_1 \oplus e_2 \ \mathsf{at}\ \rho \mid e_1 \oslash e_2 \mid$ $\mathsf{tt} \mid \mathsf{ff} \mid \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f \mid$ $x \mid \lambda x : \tau.^{\theta} e \ \mathsf{at}\ \rho \mid e_1\ e_2 \mid$ $(e_1, e_2)\ \mathsf{at}\ \rho \mid \mathsf{fst}\ e \mid \mathsf{snd}\ e \mid$ $\mathsf{letregion}\ \varrho\ \mathsf{in}\ e \mid \lambda \varrho \succeq \varphi.^{\theta} u \ \mathsf{at}\ \rho \mid e\ [\rho] \mid$ $\langle l \rangle_r \mid \langle l \rangle_{\bullet}$	
Computation abstractions	u	$::=$	$\lambda x : \tau.^{\theta} e \ \mathsf{at}\ \rho \mid \lambda \varrho \succeq \varphi.^{\theta} u \ \mathsf{at}\ \rho$	
Computation values	v	$::=$	$\mathsf{tt} \mid \mathsf{ff} \mid x \mid \langle l \rangle_r \mid \langle l \rangle_{\bullet}$	
Storable values	w	$::=$	$i \mid \lambda x : \tau.^{\theta} e \mid (v_1, v_2) \mid \lambda \varrho \succeq \varphi : \tau.^{\theta} u$	
Regions	R	$::=$	$\{l_1 \mapsto w_1, \ldots, l_n \mapsto w_n\}$	
Region stacks / Stacks	S	$::=$	$\cdot \mid S, r \mapsto R$	(ordered domain)

$$
\begin{array}{rcl}
S \supseteq_{=r} S' & \equiv & dom(S) = dom(S') \wedge \forall r \in dom(S').S(r) \supseteq_r S'(r) \\
S \supseteq_{\supseteq r} S' & \equiv & dom(S) \supseteq dom(S') \wedge \forall r \in dom(S').S(r) \supseteq_r S'(r) \\
R \supseteq_{=} R' & \equiv & dom(R) = dom(R') \wedge \forall l \in dom(R').R(l) = R'(l) \\
R \supseteq_{\supseteq} R' & \equiv & dom(R) \supseteq dom(R') \wedge \forall l \in dom(R').R(l) = R'(l)
\end{array}
$$

Figure 2: Computation syntax of SEC

2.4 Computation Syntax of SEC

Figure 2 presents the synatx of "computation programs", which extends the syntax of the previous section with semantic values that appear in the operational semantics.

Region names and locations are used to represent pointers to region allocated data. Region placeholders distinguish between live and dead stacks and regions; a dead region corresponds to a deallocated region.

The computation syntax adds no new type forms.

The computation syntax adds two new expression forms. The expression $\langle l \rangle_r$ is the runtime representation of a (μ, r); that is, it is the (live) pointer associated with a region allocated value. Likewise, the expression $\langle l \rangle_\bullet$ is the runtime representation of a $(\mu, \bullet)$; that is, it is the (dangling) pointer associated with a region deallocated value.

Thus far, we have talked about region allocated data without discussing where such data is stored. We formalize the syntactic class of storable values. Storable values are associated with locations in regions R and regions are ordered into stacks S. Intuitively, evaluating a letregion expression adds a new region to the top of the stack (which is deallocated upon finishing the expression). This intuition is formalized in the operational semantics of the next section.

Finally, we introduce relations of the form $\supseteq_{sr}$ and $\supseteq_r$, which describe a family of extensions of stacks and regions, respectively. These relations are only needed for the type-soundness proof, but we find it convenient to state their definitions along with the definitions of stacks and regions. Note that we consider stacks to have ordered domains. Hence, $dom(S) = dom(S')$ indicates that S and S' have equal ordered domains, while $dom(S) \supseteq dom(S')$ indicates that the ordered domain of $dom(S')$ is a prefix of the ordered domain of $dom(S)$.

2.5 Dynamic semantics of SEC

Two inductive judgements (Figure 3) define the dynamic semantics of the Single Effect Calculus. We state without proof that the dynamic semantics is (almost) deterministic; it is syntax-directed, but fresh region names and locations are chosen non-deterministicly.

The judgement $S; e \hookrightarrow S'; v'$ asserts that evaluating the closed expression e in stack S results in a new stack S' and a value v'. Note that the rules for $S; e \hookrightarrow S'; v'$ thread the modified stack through each expression evaluation, imposing a left-to-right evaluation order. Consider, for example, the following rule:

$$\frac{\begin{array}{c} S; e_1 \hookrightarrow S_1; \langle l_1 \rangle_{r_1} \qquad r_1 \in dom(S_1) \qquad l_1 \in dom(S_1(r_1)) \qquad S_1(r_1, l_1) = i_1 \\ S_1; e_2 \hookrightarrow S_2; \langle l_2 \rangle_{r_2} \qquad r_2 \in dom(S_2) \qquad l_2 \in dom(S_2(r_2)) \qquad S_2(r_2, l_2) = i_2 \\ r \in dom(S_2) \qquad l \notin dom(S_2(r)) \qquad i = i_1 \oplus i_2 \end{array}}{S; e_1 \oplus e_2 \text{ at } r \hookrightarrow S_2\{(r, l) \mapsto i\}; \langle l \rangle_r}$$

The first line evaluates e_1 to a live location and reads out the integer stored at $\langle l_1 \rangle_{r_1}$. Likewise, the second line evaluates e_2 to a live location and reads out the integer stored at $\langle l_2 \rangle_{r_2}$. Finally, a fresh location in the region r is chosen, and the final stack with the computed integer stored at the freshly chosen location and the location are returned. The other rules work in much the same manner.

The rule for letregion introduces (and subsequently eliminates) a new region. The rule executes in the following manner. First, a fresh region name r is chosen. Next, the region r is substituted for the region variable ϱ in the body of the letregion expression. This expression is then evaluated under an extended stack that adds an empty region bound to r to the top of the stack, yielding a modified stack and value v'', The modified top region is discarded, while occurrences of r are replaced by $\bullet$ in v'', because the region has been conceptually deallocated and is no longer accessible.

It is important to note that the execution of any expression that allocates or reads a region allocated value is predicated upon having a live region in the stack. While it will be possible to have expressions that

[1] Limiting the body of a region abstraction to abstractions ensures that an erasure function that removes region annotations and produces a λ-calculus term is meaning preserving.

$\boxed{S; e \hookrightarrow v}$

$$\frac{\rho \equiv r \qquad r \in dom(S) \qquad l \notin dom(S(r))}{S; i \text{ at } \rho \hookrightarrow S\{(r,l) \mapsto i\}; \langle l \rangle_r}$$

$$\frac{\begin{array}{c} S; e_1 \hookrightarrow S_1; v_1 \qquad v_1 \equiv \langle l_1 \rangle_{r_1} \\ r_1 \in dom(S_1) \qquad l_1 \in dom(S_1(r_1)) \qquad S_1(r_1, l_1) = i_1 \\ S_1; e_2 \hookrightarrow S_2; v_2 \qquad v_2 \equiv \langle l_2 \rangle_{r_2} \\ r_2 \in dom(S_2) \qquad l_2 \in dom(S_2(r_2)) \qquad S_2(r_2, l_2) = i_2 \\ \rho \equiv r \qquad r \in dom(S_2) \qquad l \notin dom(S_2(r)) \qquad i = i_1 \oplus i_2 \end{array}}{S; e_1 \oplus e_2 \text{ at } \rho \hookrightarrow S_2\{(r,l) \mapsto i\}; \langle l \rangle_r}$$

$$\frac{\begin{array}{c} S; e_1 \hookrightarrow S_1; v_1 \qquad v_1 \equiv \langle l_1 \rangle_{r_1} \\ r_1 \in dom(S_1) \qquad l_1 \in dom(S_1(r_1)) \qquad S_1(r_1, l_1) = i_1 \\ S_1; e_2 \hookrightarrow S_2; v_2 \qquad v_2 \equiv \langle l_2 \rangle_{r_2} \\ r_2 \in dom(S_2) \qquad l_2 \in dom(S_2(r_2)) \qquad S_2(r_2, l_2) = i_2 \\ b = i_1 \otimes i_2 \end{array}}{S; e_1 \otimes e_2 \text{ at } \varrho \hookrightarrow S_2; b}$$

$$\frac{}{S; \mathsf{tt} \hookrightarrow S; \mathsf{tt}} \qquad \frac{}{S; \mathsf{ff} \hookrightarrow S; \mathsf{ff}}$$

$$\frac{S; e_b \hookrightarrow S'; v' \qquad v' \equiv \mathsf{tt} \qquad S'; e_t \hookrightarrow S''; v''}{S; \text{if } e_b \text{ then } e_t \text{ else } e_f \hookrightarrow S''; v''}$$

$$\frac{S; e_b \hookrightarrow S'; v' \qquad v' \equiv \mathsf{ff} \qquad S'; e_f \hookrightarrow S''; v''}{S; \text{if } e_b \text{ then } e_t \text{ else } e_f \hookrightarrow S''; v''}$$

$$\frac{\rho \equiv r \qquad r \in dom(S) \qquad l \notin dom(S(r))}{S; \lambda x : \tau.^{\theta'} e' \text{ at } \rho \hookrightarrow S\{(r,l) \mapsto \lambda x : \tau.^{\theta'} e'\}; \langle l \rangle_r}$$

$$\frac{\begin{array}{c} S; e_1 \hookrightarrow S_1; v_1 \qquad v_1 \equiv \langle l_1 \rangle_{r_1} \\ r_1 \in dom(S_1) \qquad l_1 \in dom(S_1(r_1)) \qquad S_1(r_1, l_1) = \lambda x : \tau_1.^{\theta'} e' \\ S_1; e_2 \hookrightarrow S_2; v_2 \qquad S_2; e'[v_2/x] \hookrightarrow S_3; v_3 \end{array}}{S; e_1 \ e_2 \hookrightarrow S_3; v_3}$$

$$\frac{\begin{array}{c} S; e_1 \hookrightarrow S_1; v_1 \\ S_1; e_2 \hookrightarrow S_2; v_2 \\ \rho \equiv r \qquad r \in dom(S_2) \qquad l \notin dom(S_2(r)) \end{array}}{S; (e_1, e_2) \text{ at } \rho \hookrightarrow S_2\{(r,l) \mapsto (v_1, v_2)\}; \langle l \rangle_r}$$

$$\frac{\begin{array}{c} S; e \hookrightarrow S'; v \qquad v \equiv \langle l \rangle_r \\ r \in dom(S') \qquad l \in dom(S'(r)) \qquad S'(r, l) = (v_1, v_2) \end{array}}{S; \mathsf{fst}\ e \hookrightarrow S'; v_1}$$

$$\frac{\begin{array}{c} S; e \hookrightarrow S'; v \qquad v \equiv \langle l \rangle_r \\ r \in dom(S') \qquad l \in dom(S'(r)) \qquad S'(r, l) = (v_1, v_2) \end{array}}{S; \mathsf{snd}\ e \hookrightarrow S'; v_2}$$

$$\frac{r \notin dom(S) \qquad S, r \mapsto \{\}; e[r/\varrho] \hookrightarrow S'; v'' \qquad S' \equiv S'', r \mapsto R''}{S; \text{letregion } \varrho \text{ in } e \hookrightarrow S''[\bullet/r]; v''[\bullet/r]}$$

$$\frac{\rho \equiv r \qquad r \in dom(S) \qquad l \notin dom(S(r))}{S; \lambda \varrho \succeq \varphi.^{\theta'} u' \text{ at } \rho \hookrightarrow S\{(r,l) \mapsto \lambda \varrho \succeq \varphi.^{\theta'} u'\}; \langle l \rangle_r}$$

$$\frac{\begin{array}{c} S; e_1 \hookrightarrow S_1; v_1 \qquad v_1 = \langle l_1 \rangle_{r_1} \\ r_1 \in dom(S_1) \qquad l_1 \in dom(S_1(r_1)) \qquad S_1(r_1, l_1) = \lambda \varrho \succeq \varphi.^{\theta'} u' \\ S_1; u'[\rho_2/\varrho] \hookrightarrow S_2; v_2 \end{array}}{S; e_1\ [\rho_2] \hookrightarrow S_2; v_2}$$

$\boxed{p \hookrightarrow v}$

$$\frac{\cdot, \mathsf{H} \mapsto \{\}; p[\mathsf{H}/\mathcal{H}] \hookrightarrow S'; v' \qquad S' \equiv \cdot; \mathsf{H} \mapsto R'}{p \hookrightarrow v'[\bullet/\mathsf{H}]}$$

Figure 3: Dynamic semantics of SEC

Region contexts	Δ	::=	$\cdot \mid \Delta, \varrho \succeq \varphi$
Value contexts	Γ	::=	$\cdot \mid \Gamma, x : \tau$
Region domains	$\overline{\mathcal{R}}$	::=	$\{l_1, \ldots, l_n\}$
Region types	$\mathcal{R}$	::=	$\{l_1 \mapsto \mu_1, \ldots, l_n \mapsto \mu_n\}$
Region stack domains / Stack domains	$\overline{\mathcal{S}}$	::=	$\cdot \mid \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}$ (ordered domain)
Region stack types / Stack types	$\mathcal{S}$	::=	$\cdot \mid \mathcal{S}, r \mapsto \mathcal{R}$ (ordered domain)

$$
\begin{array}{lcl}
\overline{\mathcal{S}} \supseteq_{=r} \overline{\mathcal{S}}' & \equiv & dom(\overline{\mathcal{S}}) = dom(\overline{\mathcal{S}}') \wedge \forall r \in dom(\overline{\mathcal{S}}').\overline{\mathcal{S}}(r) \supseteq_r \overline{\mathcal{S}}'(r) \\
\overline{\mathcal{S}} \supseteq_{\supseteq r} \overline{\mathcal{S}}' & \equiv & dom(\overline{\mathcal{S}}) \supseteq dom(\overline{\mathcal{S}}') \wedge \forall r \in dom(\overline{\mathcal{S}}').\overline{\mathcal{S}}(r) \supseteq_r \overline{\mathcal{S}}'(r) \\
\overline{\mathcal{R}} \supseteq_{=} \overline{\mathcal{R}}' & \equiv & dom(\overline{\mathcal{R}}) = dom(\overline{\mathcal{R}}') \\
\overline{\mathcal{R}} \supseteq_{\supseteq} \overline{\mathcal{R}}' & \equiv & dom(\overline{\mathcal{R}}) \supseteq dom(\overline{\mathcal{R}}')
\end{array}
$$

Figure 4: Static semantics of SEC (definitions)

reference deallocated regions, it will not be possible to evaluate them. The type system of the next section ensures that these invariants are preserved during the execution of well-typed programs.

Finally, there is a special rule for the evaluation of surface programs. Programs are evaluated under stack with a distinguished region H, which is subtituted for the region variable $\mathcal{H}$ during the evaluation of the program. Essentially, one can consider the evaluation of a program p as being equivalent to the evaluation of the expression $\mathsf{letregion}\ \mathcal{H}\ \mathsf{in}\ p$, where the final stack is discarded.

2.6 Static Semantics of SEC

Well-typed programs obey several invariants, which are enforced with typing judgements. In addition to the traditional "type-checking" judgements for expressions, we have judgements that enforce the consistency of stacks, and various well-formedness judgements that serve as a technical convenience.

2.6.1 Definitions

Figure 4 present additional definitions for syntactic classes that appear in the static semantics. Region contexts Δ are ordered lists of region variables bounded by effect sets. Value contexts Γ are ordered lists of variables and types. These contexts record the region variables and value variables in scope. Stack and region types mimic stacks and regions, recording the (boxed) type of the value stored at each location. Stack and region domains are a technical device that records the locations in scope. Because proving the well-formedness of stack types requires proving the well-formedness of (boxed) types, which requires verifying that region names are in scope, one cannot easily have stack types serve the dual purpose of recording locations in scope.

We overload the relations of the form $\supseteq_{sr}$ and $\supseteq_r$ to describe extensions of stack and region domains and types. The same conventions described above for ordered domains apply.

The typing rules for the Single Effect Calculus appear in the following figures. We summarize the main typing judgements in the following table:

Judgement	Meaning
$\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \mu$	Boxed type μ is well-formed.
$\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau$	Type τ is well-formed.
$\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho'$	If region ρ is live, then region ρ' is live. (Alt.: region ρ' outlives region ρ.)
$\Delta; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \varphi$	If region ρ is live, then all regions in φ are live. (Alt.: all regions in φ outlive region ρ.)
$\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta$	Term e has type τ and effects bounded by region θ.
$\vdash_{\text{prog}} p\ \mathsf{ok}$	Program p is well-typed.

2.6.2 Expressions

Figure 5 present the typing rules for the judgement $\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta$, which asserts that under region context Δ, value context Γ, and stack type $\mathcal{S}$ with stack domain $\overline{\mathcal{S}}$, the expression e has type τ and effects bounded by the region θ.

Previous formulations of region calculi make use of a judgement of the form $\Gamma \vdash_{\text{exp}} e : \tau, \varphi$, where φ indicates the set of regions that may be affected by the evaluation of e (and the set of bound region variables is left implicit). The Single Effect Calculus simply replaces φ with a single region θ that bounds the effects in φ. In practice, and as suggested by the typing rules, θ usually corresponds to the most recently allocated region (also referred to as the top or current region).

We start by noting that the typing rules for the judgements $\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho'$ and $\Delta; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \varphi$ (Figure 6) simply formalize the reflexive, transitive closure of the syntactic constraints in Δ, each of which asserts a particular "outlived by" relation between a region variable and an effect set, and $\overline{\mathcal{S}}$, which asserts "outlived by" relations by explicit ordering of region names. Note that the judgement $\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho'$ is not syntax directed.

The key judgement in region calculi is the typing rule for $\mathsf{letregion}\ \varrho\ \mathsf{in}\ e$:

$$\frac{\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau \qquad \vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta \qquad \Delta, \varrho \succeq \{\theta\}; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \varrho}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \mathsf{letregion}\ \varrho\ \mathsf{in}\ e : \tau, \theta}$$

The antecedent $\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau$ asserts that the new region variable ϱ does not appear in the result type, including any effects occurring in region abstraction types that appear in the result type. Note further that the (implicit) antecedent $\varrho \notin \mathit{dom}(\Delta)$ and the (explicit) judgement $\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} : \theta$ ensure that ϱ does not appear in the types of the value environment. Together, these facts guarantee that the region ϱ is not needed before the evaluation of e, nor is it needed after, corresponding to the allocation and deallocation of a new region. This new region is clearly related to the current region θ — it is outlived by the "old" current region and becomes the "new" current region for the evaluation of e. These facts are captured by the final antecedent $\Delta, \varrho \succeq \{\theta\}; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \varrho$.

It is worth comparing the treatment of latent effects in the Single Effect Calculus with their treatment in previous formulations of region calculi. In previous work, the typing rule for application appears as follows:

$$\frac{\Gamma \vdash_{\text{exp}} e_1 : (\tau_1 \xrightarrow{\varphi} \tau_2, \rho), \varphi_1 \qquad \Gamma \vdash_{\text{exp}} e_2 : \tau_2, \varphi_2}{\Gamma \vdash_{\text{exp}} e_1\ e_2 : \tau_2, \varphi \cup \varphi_1 \cup \varphi_2 \cup \{\rho\}}$$

In the Single Effect Calculus, the composite effect $\varphi \cup \varphi_1 \cup \varphi_2 \cup \{\rho\}$ is witnessed by a single effect θ that subsumes the effect of the entire expression. We interpret θ as an upper bound on the composite effect; hence, θ is an upper bound on each of the effect sets φ_1 and φ_2, which explains why θ is used in the antecedents that type-check the sub-expressions e_1 and e_2. In order to execute the application, the operational semantics must read the function out of the region ρ; therefore, we require ρ to outlive the current region θ by the antecedent $\Delta \vdash_{\text{rr}} \theta \succeq \rho$. Finally, we require the latent single effect θ', which is an upper bound on the set of regions affected by executing the function, to outlive the current region, which ensures that θ is also an upper bound on the set of regions affected by executing the function.

As alluded to in the previous section, the typing rule for region application requires that we be able to show that the formal region parameter ρ is outlived by all of the regions in the region abstraction bound φ.

The judgements for locations ensure that region names that appear in locations are in scope; furthermore, a location in a live region points to a value with the type assigned by the stack type.

Figure 7 presents the typing rules for the judgements $\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{cval}} v : \tau$ and $\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{sto}} w : \mu$, which assert that closed and storable values are well-typed. The judgements are essentially the same as those for related terms in Figure 5. The absence of a region bounding the effect corresponds to the fact that once an expression has been evaluated to a value, it no longer gives rise to a computational effect.

$\boxed{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta}$

$$\frac{\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} i \text{ at } \rho : (\mathsf{int}, \rho), \theta}$$

$$\frac{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\mathsf{int}, \rho_1), \theta \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho_1 \qquad \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_2 : (\mathsf{int}, \rho_2), \theta \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho_2 \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 \oplus e_2 \text{ at } \rho : (\mathsf{int}, \rho), \theta}$$

$$\frac{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\mathsf{int}, \rho_1), \theta \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho_1 \qquad \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_2 : (\mathsf{int}, \rho_2), \theta \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho_2}{\Delta; \Gamma \vdash_{\text{exp}} e_1 \circledcirc e_2 : \mathsf{bool}, \theta}$$

$$\frac{\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \mathsf{tt} : \mathsf{bool}, \theta} \qquad \frac{\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \mathsf{ff} : \mathsf{bool}, \theta}$$

$$\frac{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_b : \mathsf{bool}, \theta \qquad \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_t : \tau, \theta \qquad \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_f : \tau, \theta}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f : \tau, \theta}$$

$$\frac{\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta \qquad x \in dom(\Gamma) \qquad \Gamma(x) = \tau}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} x : \tau, \theta}$$

$$\frac{\Delta; \Gamma, x : \tau_1; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e' : \tau_2, \theta' \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \lambda x : \tau_1.^{\theta'} e' \text{ at } \rho : (\tau_1 \xrightarrow{\theta'} \tau_2, \rho), \theta}$$

$$\frac{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\tau_1 \xrightarrow{\theta'} \tau_2, \rho_1'), \theta \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho_1' \qquad \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_2 : \tau_1, \theta \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \theta'}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1\ e_2 : \tau_2, \theta}$$

$$\frac{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : \tau_1, \theta \qquad \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_2 : \tau_2, \theta \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} (e_1, e_2) \text{ at } \rho : (\tau_1 \times \tau_2, \rho), \theta}$$

$$\frac{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : (\tau_1 \times \tau_2, \rho), \theta \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \mathsf{fst}\ \ e : \tau_1, \theta}$$

$$\frac{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : (\tau_1 \times \tau_2, \rho), \theta \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \mathsf{snd}\ \ e : \tau_2, \theta}$$

$$\frac{\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau \qquad \vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta \qquad \Delta, \varrho \succeq \{\theta\}; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \varrho}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \mathsf{letregion}\ \varrho\ \mathsf{in}\ e : \tau, \theta}$$

$$\frac{\Delta, \varrho \succeq \varphi; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} u' : \tau, \theta' \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \lambda \varrho \succeq \varphi.^{\theta'} u' \text{ at } \rho : (\Pi \varrho \succeq \varphi.^{\theta'} \tau, \rho), \theta}$$

$$\frac{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\Pi \varrho \succeq \varphi.^{\theta'} \tau, \rho_1'), \theta \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho_1' \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho_2 \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{re}} \rho_2 \succeq \varphi \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \theta'[\rho_2/\varrho]}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1\ [\rho_2] : \tau[\rho_2/\varrho], \theta}$$

$$\frac{\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \mu}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \langle l \rangle_\bullet : (\mu, \bullet), \theta}$$

$$\frac{\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta \qquad r \in dom(\overline{\mathcal{S}}) \qquad l \in dom(\overline{\mathcal{S}}(r)) \qquad \mu = \mathcal{S}(r, l)}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \langle l \rangle_r : (\mu, r), \theta}$$

Figure 5: Static semantics of SEC (expressions)

$\boxed{\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} \rho \succeq \rho'}$

$$\frac{\overline{\mathcal{S}} \vdash_{\mathrm{rctxt}} \Delta \qquad \Delta(\varrho) = \{\rho_1, \ldots, \rho_i, \ldots, \rho_n\}}{\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} \varrho \succeq \rho_i} \qquad \frac{\overline{\mathcal{S}} \vdash_{\mathrm{rctxt}} \Delta \qquad \overline{\mathcal{S}} = \overline{\mathcal{S}}_1, r_1 \mapsto \overline{\mathcal{R}}_1, \overline{\mathcal{S}}_2, r_2 \mapsto \overline{\mathcal{R}}_2, \overline{\mathcal{S}}_3}{\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r_2 \succeq r_1} \qquad \frac{\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{re}} r \succeq \{r_1, \ldots, r_n\}}{\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r \succeq r_i}$$

$$\frac{\overline{\mathcal{S}} \vdash_{\mathrm{rctxt}} \Delta \qquad \overline{\mathcal{S}} = \overline{\mathcal{S}}_1, r_1 \mapsto R_1, \overline{\mathcal{S}}_2}{\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} \bullet \succeq r_1} \qquad \frac{\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{place}} \rho}{\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} \rho \succeq \rho} \qquad \frac{\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} \rho \succeq \rho' \qquad \Delta; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} \rho' \succeq \rho''}{\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} \rho \succeq \rho''}$$

$\boxed{\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{re}} \rho \succeq \varphi}$

$$\frac{\overline{\mathcal{S}} \vdash_{\mathrm{rctxt}} \Delta \qquad \Delta; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} \rho \succeq \rho_i \quad {}^{i \in 1 \ldots n}}{\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{re}} \rho \succeq \{\rho_1, \ldots, \rho_n\}}$$

Figure 6: Static semantics of SEC (casts)

$\boxed{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\mathrm{cval}} v : \tau}$

$$\frac{\vdash_{\mathrm{stype}} \mathcal{S} : \overline{\mathcal{S}}}{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\mathrm{cval}} \mathsf{tt} : \mathsf{bool}} \qquad \frac{\vdash_{\mathrm{stype}} \mathcal{S} : \overline{\mathcal{S}}}{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\mathrm{cval}} \mathsf{ff} : \mathsf{bool}} \qquad \frac{\vdash_{\mathrm{stype}} \mathcal{S} : \overline{\mathcal{S}} \qquad \cdot; \overline{\mathcal{S}} \vdash_{\mathrm{btype}} \mu}{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\mathrm{cval}} \langle l \rangle_{\bullet} : (\mu, \bullet)}$$

$$\frac{\vdash_{\mathrm{stype}} \mathcal{S} : \overline{\mathcal{S}} \qquad r \in dom(\overline{\mathcal{S}}) \qquad l \in dom(\overline{\mathcal{S}}(r)) \qquad \mu = \mathcal{S}(r, l)}{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\mathrm{cval}} \langle l \rangle_r : (\mu, r)}$$

$\boxed{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\mathrm{sto}} w : \mu}$

$$\frac{\vdash_{\mathrm{stype}} \mathcal{S} : \overline{\mathcal{S}}}{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\mathrm{sto}} i : \mathsf{int}} \qquad \frac{\cdot; \cdot, x : \tau_1; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\mathrm{exp}} e' : \tau_2, \theta'}{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\mathrm{sto}} \lambda x : \tau_1.^{\theta'} e' : \tau_1 \xrightarrow{\theta'} \tau_2} \qquad \frac{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\mathrm{cval}} v_1 : \tau_1 \qquad \mathcal{S} : \overline{\mathcal{S}} \vdash_{\mathrm{cval}} v_2 : \tau_2}{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\mathrm{val}} (v_1, v_2) : \tau_1 \times \tau_2}$$

$$\frac{\cdot, \varrho \succeq \varphi; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\mathrm{exp}} u' : \tau, \theta'}{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\mathrm{sto}} \lambda \varrho \succeq \varphi.^{\theta'} u' : \Pi \varrho \succeq \varphi.^{\theta'} \tau}$$

Figure 7: Static semantics of SEC (closed and storable values)

$\boxed{\vdash_{\text{sdom}} \overline{\mathcal{S}}}$

$$\frac{}{\vdash_{\text{sdom}} \cdot} \qquad \frac{\vdash_{\text{sdom}} \overline{\mathcal{S}} \qquad r \notin dom(\overline{\mathcal{S}}) \qquad \vdash_{\text{rdom}} \overline{\mathcal{R}}}{\vdash_{\text{sdom}} \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}}$$

$\boxed{\vdash_{\text{rdom}} \overline{\mathcal{R}}}$

$$\frac{i \neq j \Rightarrow l_i \neq l_j \ ^{i \in 1 \ldots n, j \in 1 \ldots n}}{\vdash_{\text{rdom}} \{l_1, \ldots, l_n\}}$$

$\boxed{\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}}$

$$\frac{\begin{array}{c}\vdash_{\text{sdom}} \overline{\mathcal{S}} \\ dom(\overline{\mathcal{S}}) = dom(\mathcal{S}) \\ \forall r \in dom(\overline{\mathcal{S}}).\ dom(\overline{\mathcal{S}}(r)) = dom(\mathcal{S}(r)) \\ \forall r \in dom(\overline{\mathcal{S}}). \forall l \in dom(\overline{\mathcal{S}}(r)).\ \cdot; \overline{\mathcal{S}} \vdash_{\text{btype}} \mathcal{S}(r, l)\end{array}}{\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}}$$

$\boxed{\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}}$

$$\frac{\begin{array}{c}\vdash_{\text{sdom}} \overline{\mathcal{S}} \\ \vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}} \\ dom(\overline{\mathcal{S}}) = dom(\mathcal{S}) = dom(S) \\ \forall r \in dom(\overline{\mathcal{S}}).\ dom(\overline{\mathcal{S}}(r)) = dom(\mathcal{S}(r)) = dom(S(r)) \\ \forall r \in dom(\overline{\mathcal{S}}). \forall l \in dom(\overline{\mathcal{S}}(r)).\ \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{sto}} S(r, l) : (\mathcal{S}(r, l), r)\end{array}}{\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}}$$

Figure 8: Static semantics of SEC (stacks and regions)

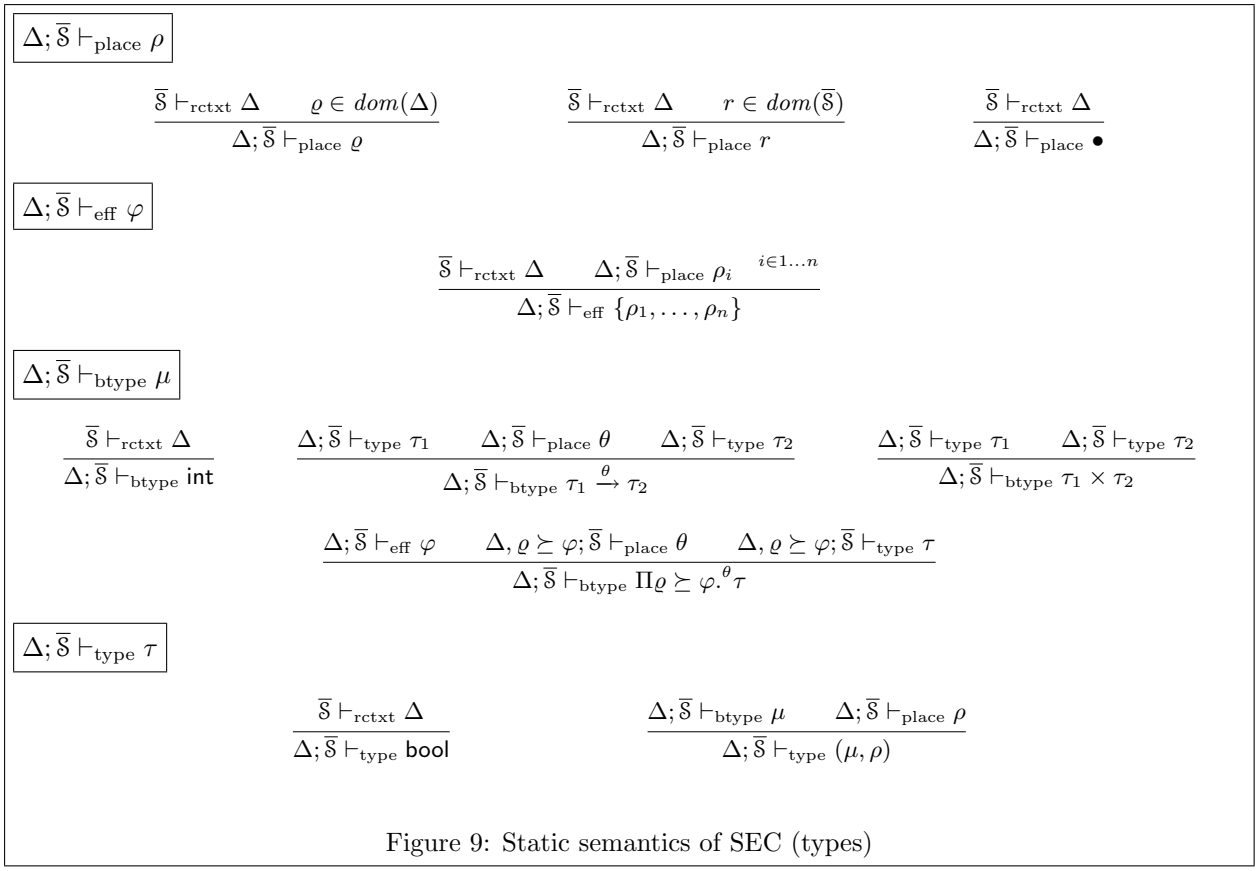


Figure 9: Static semantics of SEC (types)

2.6.3 Stacks

Figure 8 present typing rules that enforce the well-formedness and consistency of stacks. The judgements $\vdash_{\text{sdom}}$ and $\vdash_{\text{rdom}}$ simply require that stack and region domains contain distinct region names and locations. The judgement $\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}$ asserts that stack type $\mathcal{S}$ is well-formed with tower domain $\overline{\mathcal{S}}$. In particular, the judgement asserts that $\mathcal{S}$ has the domain specified by $\overline{\mathcal{S}}$ and each (boxed) type in the range of $\mathcal{S}$ is well-formed. Note that the judgement is made with respect to the entire stack domain $\overline{\mathcal{S}}$. This allows types "lower" in the stack can reference region names that appear "higher" in the tower. This corresponds to the fact that one can have arbitrary pointers between region allocated data. Finally, the judgement $\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}$ asserts that the tower S is well-formed with tower type $\mathcal{S}$ and tower domain $\overline{\mathcal{S}}$. Like the judgement $\vdash_{\text{stype}}$, it asserts that S has the domain specified by $\overline{\mathcal{S}}$ and each stored value in the range of S has the type specified by $\mathcal{S}\mathcal{T}$.

2.6.4 Technical details

Figures 9 and 10 contain additional judgements for ensuring that types τ, region contexts Δ, and value contexts Γ are well-formed. Because effect sets and types may contain region names, the judgements $\vdash_{\text{btype}}$, $\vdash_{\text{type}}$, $\vdash_{\text{rctxt}}$, and $\vdash_{\text{vctxt}}$ require a stack domain $\overline{\mathcal{S}}$.

2.6.5 Surface programs

Since surface programs do not admit syntax for naming regions, we adopt the judgement $\vdash_{\text{prog}} p \ \mathsf{ok}$ given in Figure 11. The rule for top-level surface programs requires that an expression evaluate to a boolean value in the context of distinguished region $\mathcal{H}$ that remains live throughout the execution of the program. It also

$\boxed{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta}$

$$\frac{\vdash_{\text{sdom}} \overline{\mathcal{S}}}{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \cdot} \qquad \frac{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta \qquad \varrho \notin dom(\Delta) \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{eff}} \varphi}{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta, \varrho \succeq \varphi}$$

$\boxed{\Delta; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \Gamma}$

$$\frac{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta}{\Delta; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \cdot} \qquad \frac{\Delta; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \Gamma \qquad x \notin dom(\Gamma) \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau}{\Delta; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \Gamma, x : \tau}$$

$\boxed{\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta}$

$$\frac{\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \Gamma}{\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta}$$

Figure 10: Static semantics of SEC (contexts)

$\boxed{\vdash_{\text{prog}} p \ \mathsf{ok}}$

$$\frac{\cdot, \mathcal{H} \succeq \{\}; \cdot; \cdot : \cdot \vdash_{\text{exp}} p : \mathsf{bool}, \mathcal{H}}{\vdash_{\text{prog}} p \ \mathsf{ok}}$$

Figure 11: Static semantics of SEC (surface programs)

serves as the single effect that bounds the effects of the entire program. Alternative formulations of these "boundary conditions" exist; we have adopted these to simplify the translation in Section 4.

It is useful to note that the static semantics can be greatly simplified given this rule for surface programs. Stack types and stack domains are purely technical devices that are used to prove type soundness. In the static semantics, they simply collect the names of regions in scope and assign types to locations. Note that in every rule, stack types and stack domains are passed unmodified to sub-judgements. Pushing these empty stack types and stack domains through the rules leads to the following simplifications:

$\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta$	$\Longrightarrow$	$\Delta; \Gamma \vdash_{\text{exp}} e : \tau, \theta$
$\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau$	$\Longrightarrow$	$\Delta \vdash_{\text{btype}} \tau$
$\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \mu$	$\Longrightarrow$	$\Delta \vdash_{\text{btype}} \mu$
$\Delta; \overline{\mathcal{S}} \vdash_{\text{eff}} \varphi$	$\Longrightarrow$	$\Delta \vdash_{\text{eff}} \varphi$
$\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho$	$\Longrightarrow$	$\Delta \vdash_{\text{place}} \rho$
$\Delta; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \varphi$	$\Longrightarrow$	$\Delta \vdash_{\text{re}} \rho \succeq \varphi$
$\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho'$	$\Longrightarrow$	$\Delta \vdash_{\text{rr}} \rho \succeq \rho'$
$\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta$	$\Longrightarrow$	$\vdash_{\text{rctxt}} \Delta$
$\Delta; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \Gamma$	$\Longrightarrow$	$\Delta \vdash_{\text{vctxt}} \Gamma$
$\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta$	$\Longrightarrow$	$\vdash_{\text{ctxt}} \Delta; \Gamma; \theta$

2.7 Type Soundness of SEC

In this section, we prove type soundness. The lemmas are presented in "bottom-up" order, where all relevant lemmas are presented (and proved) before being used. The lemmas follow the conventional structure of a syntactic type-soundness arguments. However, we first give a "top-down" overview of the argument.

Since our utimate goal is to demonstrate a type- and semantics-preserving translation of the Single Effect Calculus into $\mathsf{F}^{\mathsf{RGN}}$, we forgoe proving a Progress Theorem (such a proof would be very similar to the Progress Theorem for $\mathsf{F}^{\mathsf{RGN}}$ given in Section 3.5).

The Preservation Theorem states that the terminating computation of a well-typed expression yields a well-typed extension of the stack and a (closed) value of the same type. The various substitution lemmas for dead regions are required to prove the cases where regions are deallocated.

The various Well-Formedness Lemmas are useful technical facts that alleviate the need to assume contexts are well-formed.

2.7.1 Lemmas

Lemma 1 (Well-Formedness (from $\vdash_{\text{rctxt}}$))

If $\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta$,
then $\vdash_{\text{sdom}} \overline{\mathcal{S}}$.

Proof. By induction on the derivation $\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta$. □

Lemma 2 (Well-Formedness (from $\vdash_{\text{place}}$))

If $\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho$,
then $\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta$.

Proof. By inspection of the derivation $\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho$. □

Lemma 3 (Well-Formedness (from $\vdash_{\text{eff}}$))

If $\Delta; \overline{\mathcal{S}} \vdash_{\text{eff}} \varphi$,
then $\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta$.

Proof. By inspection of the derivation $\Delta; \overline{\mathcal{S}} \vdash_{\text{eff}} \varphi$. □

Lemma 4 (Well-Formedness (from $\vdash_{\text{btype}}$ and $\vdash_{\text{type}}$))

(1) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \mu$,
then $\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta$.

(2) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau$,
then $\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta$.

Proof. By mutual induction on the derivations $\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \mu$ and $\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau$. □

Lemma 5 (Well-Formedness (from $\vdash_{\text{vctxt}}$))

If $\Delta; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \Gamma$,
then $\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta$.

Proof. By induction on the derivation $\Delta; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \Gamma$. □

Lemma 6 (Well-Formedness (from $\vdash_{\text{ctxt}}$))

If $\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta$,
then $\vdash_{\text{sdom}} \overline{\mathcal{S}}$, $\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}$, $\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta$, $\Delta; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \Gamma$, *and* $\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \theta$.

Proof. By inspection of the derivation $\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta$. □

Lemma 7 (Well-Formedness (from $\vdash_{\text{rr}}$ and $\vdash_{\text{re}}$))

(1) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho'$,
then $\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho$ *and* $\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho'$.

(2) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \varphi$,
then $\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho$ *and* $\Delta; \overline{\mathcal{S}} \vdash_{\text{eff}} \varphi$.

Proof. By mutual induction on the derivations $\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho'$ and $\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \varphi$. □

Lemma 8 (Substitution (places))

(1) If $\Delta, \varrho \succeq \varphi, \Delta'; \overline{\mathcal{S}} \vdash_{\text{place}} \rho'$ *and* $\Delta, \Delta'[\rho/\varrho]; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \varphi$,
then $\Delta, \Delta'[\rho/\varrho]; \overline{\mathcal{S}} \vdash_{\text{place}} \rho'[\rho/\varrho]$.

(2) If $\Delta, \varrho \succeq \varphi, \Delta'; \overline{\mathcal{S}} \vdash_{\text{eff}} \varphi'$ *and* $\Delta, \Delta'[\rho/\varrho]; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \varphi$,
then $\Delta, \Delta'[\rho/\varrho]; \overline{\mathcal{S}} \vdash_{\text{eff}} \varphi'[\rho/\varrho]$.

(3) If $\Delta, \varrho \succeq \varphi, \Delta'; \overline{\mathcal{S}} \vdash_{\text{btype}} \mu$ *and* $\Delta, \Delta'[\rho/\varrho]; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \varphi$,
then $\Delta, \Delta'[\rho/\varrho]; \overline{\mathcal{S}} \vdash_{\text{btype}} \mu[\rho/\varrho]$.

(4) If $\Delta, \varrho \succeq \varphi, \Delta'; \overline{\mathcal{S}} \vdash_{\text{type}} \tau$ *and* $\Delta, \Delta'[\rho/\varrho]; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \varphi$,
then $\Delta, \Delta'[\rho/\varrho]; \overline{\mathcal{S}} \vdash_{\text{type}} \tau[\rho/\varrho]$.

Proof. By (mutual) induction on the derivations. □

Lemma 9 (Well-Formedness (from $\vdash_{\text{exp}}$))

If $\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta$,
then $\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta$ *and* $\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau$.

Proof. By induction on the derivation $\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta$. □

Lemma 10 (Substitution (places in $\vdash_{\text{exp}}$))

If $\Delta, \varrho \succeq \varphi, \Delta'; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta$ *and* $\Delta, \Delta'[\rho/\varrho]; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \varphi$,
then $\Delta, \Delta'[\rho/\varrho]; \Gamma[\rho/\varrho]; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e[\rho/\varrho] : \tau[\rho/\varrho], \theta[\rho/\varrho]$.

Proof. By induction on the derivation of $\Delta, \varrho \succeq \varphi, \Delta'; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta$ □

Lemma 11 (Substitution (closed values in $\vdash_{\text{exp}}$))

If $\Delta; \Gamma, x : \tau', \Gamma'; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta$ *and* $\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{cval}} v : \tau'$,
then $\Delta; \Gamma, \Gamma'; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e[v/x] : \tau, \theta$.

Proof. By induction on the derivation $\Delta; \Gamma, x : \tau', \Gamma'; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta$. □

Lemma 12 (Region Context Weakening)

Suppose $\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta, \Delta'$.

(1) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho$,
then $\Delta, \Delta'; \overline{\mathcal{S}} \vdash_{\text{place}} \rho$.

(2) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho$, *then* $\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho$.

(3) If $\Delta, \Delta'; \overline{\mathcal{S}} \vdash_{\text{eff}} \varphi$, *then* $\Delta, \Delta'; \overline{\mathcal{S}} \vdash_{\text{eff}} \varphi$.

(4) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho'$, *then* $\Delta, \Delta'; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho'$.

(5) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \varphi$, *then* $\Delta, \Delta'; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \varphi$.

(6) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \mu$, *then* $\Delta, \Delta'; \overline{\mathcal{S}} \vdash_{\text{btype}} \mu$.

(7) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau$, *then* $\Delta, \Delta'; \overline{\mathcal{S}} \vdash_{\text{type}} \tau$.

(8) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \Gamma$, *then* $\Delta, \Delta'; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \Gamma$.

(9) If $\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta$,
then $\vdash_{\text{ctxt}} \Delta, \Delta'; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta$.

(10) If $\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta$,
then $\Delta, \Delta'; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta$.

Proof. By (mutual) induction on the derivations. □

Lemma 13 (Stack Domain Weakening)

Suppose $\overline{\mathcal{S}}' \supseteq_{sr} \overline{\mathcal{S}}$ *and* $\vdash_{\text{sdom}} \overline{\mathcal{S}'}$.

(1) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho$, *then* $\Delta; \overline{\mathcal{S}}' \vdash_{\text{place}} \rho$.

(2) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{eff}} \varphi$, *then* $\Delta; \overline{\mathcal{S}}' \vdash_{\text{eff}} \varphi$.

(3) If $\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta$, *then* $\overline{\mathcal{S}}' \vdash_{\text{rctxt}} \Delta$.

(4) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho'$, *then* $\Delta; \overline{\mathcal{S}}' \vdash_{\text{rr}} \rho \succeq \rho'$.

(5) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \varphi$, *then* $\Delta; \overline{\mathcal{S}}' \vdash_{\text{re}} \rho \succeq \varphi$.

(6) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \mu$, *then* $\Delta; \overline{\mathcal{S}}' \vdash_{\text{btype}} \mu$.

(7) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau$, *then* $\Delta; \overline{\mathcal{S}}' \vdash_{\text{type}} \tau$.

(8) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \Gamma$, *then* $\Delta; \overline{\mathcal{S}}' \vdash_{\text{vctxt}} \Gamma$.

Further suppose $\mathcal{S}' \supseteq_{sr} \mathcal{S}$ *and* $\vdash_{\text{stype}} \mathcal{S}' : \overline{\mathcal{S}}'$.

(9) If $\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta$,
then $\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S}' : \overline{\mathcal{S}}'; \theta$.

(10) If $\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta$,
then $\Delta; \Gamma; \mathcal{S}' : \overline{\mathcal{S}}' \vdash_{\text{exp}} e : \tau, \theta$.

(11) If $\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{cval}} v : \tau$,
then $\mathcal{S}' : \overline{\mathcal{S}}' \vdash_{\text{cval}} v : \tau$.

(12) If $\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{sto}} w : \mu$,
then $\mathcal{S}' : \overline{\mathcal{S}}' \vdash_{\text{sto}} w : \mu$.

Proof. By (mutual) induction on the derivations. □

Lemma 14 (Substitution (•))

(1) *If* $\vdash_{\text{dom}} \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}$,
then $\vdash_{\text{sdom}} \overline{\mathcal{S}}$.

(2) *If* $\Delta; \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{place}} \rho$,
then $\Delta[\bullet/r]; \overline{\mathcal{S}}[\bullet/r] \vdash_{\text{place}} \rho[\bullet/r]$.

(3) *If* $\Delta; \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{eff}} \varphi$,
then $\Delta[\bullet/r]; \overline{\mathcal{S}}[\bullet/r] \vdash_{\text{eff}} \varphi[\bullet/r]$.

(4) *If* $\overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{rctxt}} \Delta$,
then $\overline{\mathcal{S}}[\bullet/r] \vdash_{\text{rctxt}} \Delta[\bullet/r]$.

(5) *If* $\Delta; \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{rr}} \rho \succeq \rho'$,
then $\Delta[\bullet/r]; \overline{\mathcal{S}}[\bullet/r] \vdash_{\text{rr}} \rho[\bullet/r] \succeq \rho'[\bullet/r]$.

(6) *If* $\Delta; \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{re}} \rho \succeq \varphi$,
then $\Delta[\bullet/r]; \overline{\mathcal{S}}[\bullet/r] \vdash_{\text{re}} \rho[\bullet/r] \succeq \varphi[\bullet/r]$.

(7) *If* $\Delta; \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{btype}} \mu$,
then $\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \mu[\bullet/r]$.

(8) *If* $\Delta; \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{type}} \tau$,
then $\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau[\bullet/r]$.

(9) *If* $\vdash_{\text{stype}} \mathcal{S}, r \mapsto \mathcal{R} : \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}$,
then $\vdash_{\text{stype}} \mathcal{S}[\bullet/r] : \overline{\mathcal{S}}$.

(10) *If* $\Delta; \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{vctxt}} \Gamma$,
then $\Delta[\bullet/r]; \overline{\mathcal{S}}[\bullet/r] \vdash_{\text{vctxt}} \Gamma[\bullet/r]$.

(11) *If* $\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S}, r \mapsto \mathcal{R} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}$,
then $\vdash_{\text{ctxt}} \Delta; \Gamma[s\sharp \bullet / s\sharp r]; \mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet / s\sharp r] : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$.

(12) *If* $\Delta; \Gamma; \mathcal{S}, r \mapsto \mathcal{R} : \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{exp}} e : \tau, \theta$,
then $\Delta[\bullet/r]; \Gamma[\bullet/r]; \mathcal{S}[\bullet/r] : \overline{\mathcal{S}} \vdash_{\text{exp}} e[\bullet/r] : \tau[\bullet/r], \theta[\bullet/r]$.

(13) *If* $\mathcal{S}, r \mapsto \mathcal{R} : \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{exp}} v : \tau$,
then $\mathcal{S}[\bullet/r] : \overline{\mathcal{S}} \vdash_{\text{exp}} v[\bullet/r] : \tau[\bullet/r]$.

(14) *If* $\mathcal{S}, r \mapsto \mathcal{R} : \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{sto}} w : \mu$,
then $\mathcal{S}[\bullet/r] : \overline{\mathcal{S}} \vdash_{\text{sto}} w[\bullet/r] : \mu[\bullet/r]$.

(15) *If* $\vdash_{\text{stack}} S, r \mapsto R : \mathcal{S}, r \mapsto \mathcal{R} : \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}$,
then $\vdash_{\text{stack}} S[\bullet/r] : \mathcal{S}[\bullet/r] : \overline{\mathcal{S}}$.

Proof. By (mutual) induction on the derivations. □

Lemma 15 (Useless Substitution (ϱ))

Suppose $\varrho \notin dom(\Delta)$ *and* $\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho'$.

(1) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho$,
then $\rho[\rho'/\varrho] \equiv \rho$.

(2) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{eff}} \varphi$,
then $\varphi[\rho'/\varrho] \equiv \varphi$.

(3) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \mu$,
then $\mu[\rho'/\varrho] \equiv \mu$.

(4) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau$,
then $\tau[\rho'/\varrho] \equiv \tau$.

Proof. By (mutual) induction on the derivations. □

Lemma 16 (Useless Substitution ($\bullet$))

Suppose $r \notin dom(\overline{\mathcal{S}})$.

(1) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho$,
then $\rho[\bullet/r] \equiv \rho$.

(2) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{eff}} \varphi$,
then $\varphi[\bullet/r] \equiv \varphi$.

(3) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \mu$,
then $\mu[\bullet/r] \equiv \mu$.

(4) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau$,
then $\tau[\bullet/r] \equiv \tau$.

Proof. By (mutual) induction on the derivations. □

Lemma 17 (Region Bound)

(1) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} r \succeq \rho'$,
then $\rho' \equiv r'$.

(2) If $\Delta; \overline{\mathcal{S}} \vdash_{\text{re}} r \succeq \varphi$,
then $\varphi \equiv \{r_1, \ldots, r_n\}$.

Proof. By mutual induction on the derivations $\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} r \succeq \rho'$ and $\Delta; \overline{\mathcal{S}} \vdash_{\text{re}} r \succeq \varphi$. □

Lemma 18 (Canonical Forms)

If $\cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} v : \tau, \theta$, *then* $\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{cval}} v : \tau$.
Furthermore

- *if* $\tau \equiv \mathsf{bool}$, *then* $v \equiv \mathsf{tt}$ *or* $v \equiv \mathsf{ff}$.
- *if* $\tau \equiv (\mu, \rho)$, *then* $v \equiv \langle l \rangle_\rho$.

Proof. Immediate by inspection of the typing rules. □

2.7.2 Preservation

Theorem 1 (Preservation)

(1) If

(a) $\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}$,

(b) $\cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, r'$, *and*

(c) $S; e \hookrightarrow S'; v'$,

then there exists $\overline{\mathcal{S}}' \supseteq_{=\supseteq} \overline{\mathcal{S}}$ *and* $\mathcal{S}' \supseteq_{=\supseteq} \mathcal{S}$ *such that* $\vdash_{\text{stack}} S' : \mathcal{S}' : \overline{\mathcal{S}}'$ *and* $\mathcal{S}' : \overline{\mathcal{S}}' \vdash_{\text{cval}} v' : \tau$.

Proof. Proceed by induction on the derivation (1c) $S; e \hookrightarrow S'; v'$.

Case $\dfrac{\rho \equiv r \qquad r \in dom(S) \qquad l \notin dom(S(r))}{S; i \text{ at } \rho \hookrightarrow S\{(r,l) \mapsto i\}; \langle l \rangle_r}$: By inspection, the derivation of (1b) must end with

$$\dfrac{\vdash_{\text{ctxt}} \cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}}; r' \qquad \cdot;\overline{\mathcal{S}} \vdash_{\text{place}} r \qquad \cdot;\overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq r}{\cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} i \text{ at } r : (\mathsf{int}, r), r'}$$

Note that $\overline{\mathcal{S}}\{(r,l) \mapsto\} \supseteq_{=\supseteq} \overline{\mathcal{S}}$ and $\mathcal{S}\{(r,l) \mapsto \mathsf{int}\} \supseteq_{=\supseteq} \mathcal{S}$. Note that $\vdash_{\text{stack}} S\{(r,l) \mapsto i\} : \mathcal{S}\{(r,l) \mapsto \mathsf{int}\} : \overline{\mathcal{S}}\{(r,l) \mapsto\}$ (requires appealing to Lemma 13). Note that

$$\dfrac{\boxed{r \in dom(\overline{\mathcal{S}}\{(r,l) \mapsto\})} \qquad \boxed{\vdash_{\text{stype}} \mathcal{S}\{(r,l) \mapsto \mathsf{int}\} : \overline{\mathcal{S}}\{(r,l) \mapsto\}} \qquad \boxed{l \in dom((\overline{\mathcal{S}}\{(r,l) \mapsto\})(r))} \qquad \boxed{\mathsf{int} = (\mathcal{S}\{(r,l) \mapsto \mathsf{int}\})(r,l)}}{\mathcal{S}\{(r,l) \mapsto \mathsf{int}\} : \overline{\mathcal{S}}\{(r,l) \mapsto\} \vdash_{\text{cval}} \langle l \rangle_r : (\mathsf{int}, r)}$$

Hence, $\overline{\mathcal{S}}\{(r,l) \mapsto\} \supseteq_{=\supseteq} \overline{\mathcal{S}}$, $\mathcal{S}\{(r,l) \mapsto \mathsf{int}\} \supseteq_{=\supseteq} \mathcal{S}$, $\vdash_{\text{stack}} S\{(r,l) \mapsto i\} : \mathcal{S}\{(r,l) \mapsto \mathsf{int}\} : \overline{\mathcal{S}}\{(r,l) \mapsto \}$, and $\mathcal{S}\{(r,l) \mapsto \mathsf{int}\} : \overline{\mathcal{S}}\{(r,l) \mapsto\} \vdash_{\text{cval}} \langle l \rangle_r : (\mathsf{int}, r)$, as required.

Case $\dfrac{\begin{array}{c} S; e_1 \hookrightarrow S_1; v_1 \qquad v_1 \equiv \langle l_1 \rangle_{r_1} \\ r_1 \in dom(S_1) \qquad l_1 \in dom(S_1(r_1)) \qquad S_1(r_1, l_1) = i_1 \\ S_1; e_2 \hookrightarrow S_2; v_2 \qquad v_2 \equiv \langle l_2 \rangle_{r_2} \\ r_2 \in dom(S_2) \qquad l_2 \in dom(S_2(r_2)) \qquad S_2(r_2, l_2) = i_2 \end{array} \qquad \rho \equiv r \qquad r \in dom(S_2) \qquad l \notin dom(S_2(r)) \qquad i = i_1 \oplus i_2}{S; e_1 \oplus e_2 \text{ at } \rho \hookrightarrow S_2\{(r,l) \mapsto i\}; \langle l \rangle_r}$: By inspection, the derivation of (1b) must end with

$$\dfrac{\begin{array}{c} \cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\mathsf{int}, \rho_1), r' \qquad \cdot;\overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq \rho_1 \\ \cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_2 : (\mathsf{int}, \rho_2), r' \qquad \cdot;\overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq \rho_2 \\ \cdot;\overline{\mathcal{S}} \vdash_{\text{place}} r \qquad \cdot;\overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq r \end{array}}{\cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 \oplus e_2 \text{ at } r : (\mathsf{int}, r), r'}$$

Applying the induction hypothesis to (1a) $\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}$, (1b) $\cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\mathsf{int}, \rho_1), r'$, and (1c) $S; e_1 \hookrightarrow S_1; \langle l_1 \rangle_{r_1}$, we conclude that there exists $\overline{\mathcal{S}}_1 \supseteq_{=\supseteq} \overline{\mathcal{S}}$ and $\mathcal{S}_1 \supseteq_{=\supseteq} \mathcal{S}$ such that $\vdash_{\text{stack}} S_1 : \mathcal{S}_1 : \overline{\mathcal{S}}_1$ and $\mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{cval}} \langle l_1 \rangle_{r_1} : (\mathsf{int}, \rho_1)$.

By Lemma 13, we conclude $\cdot;\cdot;\mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{exp}} e_2 : (\mathsf{int}, \rho_2), r'$.

Applying the induction hypothesis to (1a) $\vdash_{\text{stack}} S_1 : \mathcal{S}_1 : \overline{\mathcal{S}}_1$, (1b) $\cdot;\cdot;\mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{exp}} e_2 : (\mathsf{int}, \rho_2), r'$, and (1c) $S_1; e_2 \hookrightarrow S_2; \langle l_2 \rangle_{r_2}$, we conclude that there exists $\overline{\mathcal{S}}_2 \supseteq_{=\supseteq} \overline{\mathcal{S}}_1$ and $\mathcal{S}_2 \supseteq_{=\supseteq} \mathcal{S}_1$ such that $\vdash_{\text{stack}} S_2 : \mathcal{S}_2 : \overline{\mathcal{S}}_2$ and $\mathcal{S}_2 : \overline{\mathcal{S}}_2 \vdash_{\text{cval}} \langle l_2 \rangle_{r_2} : (\mathsf{int}, \rho_2)$.

Note that $\overline{\mathcal{S}}_2\{(r,l) \mapsto\} \supseteq_{=\supseteq} \overline{\mathcal{S}}_2$ and $\mathcal{S}_2\{(r,l) \mapsto \mathsf{int}\} \supseteq_{=\supseteq} \mathcal{S}_2$. Note that $\vdash_{\text{stack}} S_2\{(r,l) \mapsto i\} : \mathcal{S}_2\{(r,l) \mapsto \mathsf{int}\} : \overline{\mathcal{S}}_2\{(r,l) \mapsto\}$ (requires appealing to Lemma 13).

Note that

$$\dfrac{\boxed{r \in dom(\overline{\mathcal{S}}_2\{(r,l) \mapsto\})} \qquad \boxed{\vdash_{\text{stype}} \mathcal{S}_2\{(r,l) \mapsto \mathsf{int}\} : \overline{\mathcal{S}}_2\{(r,l) \mapsto\}} \qquad \boxed{l \in dom((\overline{\mathcal{S}}_2\{(r,l) \mapsto\})(r))} \qquad \boxed{\mathsf{int} = (\mathcal{S}_2\{(r,l) \mapsto \mathsf{int}\})(r,l)}}{\mathcal{S}_2\{(r,l) \mapsto \mathsf{int}\} : \overline{\mathcal{S}}_2\{(r,l) \mapsto\} \vdash_{\text{cval}} \langle l \rangle_r : (\mathsf{int}, r)}$$

By transitivity of $\supseteq_{=\supseteq}$, we conclude $\overline{\mathcal{S}}_2\{(r,l) \mapsto\} \supseteq_{=\supseteq} \overline{\mathcal{S}}$, $\mathcal{S}_2\{(r,l) \mapsto \mathsf{int}\} \supseteq_{=\supseteq} \mathcal{S}$, $\vdash_{\text{stack}} S_2\{(r,l) \mapsto i\} : \mathcal{S}_2\{(r,l) \mapsto \mathsf{int}\} : \overline{\mathcal{S}}_2\{(r,l) \mapsto\}$, and $\mathcal{S}_2\{(r,l) \mapsto \mathsf{int}\} : \overline{\mathcal{S}}_2\{(r,l) \mapsto\} \vdash_{\text{cval}} \langle l \rangle_r : (\mathsf{int}, r)$, as required.

Case $\dfrac{\begin{array}{c} S; e_1 \hookrightarrow S_1; v_1 \qquad v_1 \equiv \langle l_1 \rangle_{r_1} \\ r_1 \in dom(S_1) \qquad l_1 \in dom(S_1(r_1)) \qquad S_1(r_1, l_1) = i_1 \\ S_1; e_2 \hookrightarrow S_2; v_2 \qquad v_2 \equiv \langle l_2 \rangle_{r_2} \\ r_2 \in dom(S_2) \qquad l_2 \in dom(S_2(r_2)) \qquad S_2(r_2, l_2) = i_2 \\ b = i_1 \otimes i_2 \end{array}}{S; e_1 \otimes e_2 \text{ at } \varrho \hookrightarrow S_2; b}$: ...

Case $\dfrac{}{S; \mathsf{tt} \hookrightarrow S; \mathsf{tt}}$: By inspection, the derivation of (1b) must end with

$$\dfrac{\vdash_{\text{ctxt}} \cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}}; r'}{\cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \mathsf{tt} : \mathsf{bool}, r'}$$

Note that

$$\dfrac{\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}; r'}{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{cval}} \mathsf{tt} : \mathsf{bool}}$$

We conclude $\overline{\mathcal{S}} \supseteq_{=\supseteq} \overline{\mathcal{S}}$, $\mathcal{S} \supseteq_{=\supseteq} \mathcal{S}$, and $\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}$ and $\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{cval}} \mathsf{tt} : \mathsf{bool}$, as required.

Case $\dfrac{}{S; \mathsf{ff} \hookrightarrow S; \mathsf{ff}}$: ...

Case $\dfrac{S; e_b \hookrightarrow S'; v' \qquad v' \equiv \mathsf{tt} \qquad S'; e_t \hookrightarrow S''; v''}{S; \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f \hookrightarrow S''; v''}$: ... By inspection, the derivation of (1b) must end with

$$\dfrac{\begin{array}{c} \cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_b : \mathsf{bool}, r \\ \cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_t : \tau, r' \qquad \cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_f : \tau, r' \end{array}}{\cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f : \tau, r'}$$

Applying the induction hypothesis to (1a) $\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}$, (1b) $\cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_b : \mathsf{bool}, r'$, and (1c) $S; e_1 \hookrightarrow S_1; \mathsf{tt}$, we conclude that there exists $\overline{\mathcal{S}}' \supseteq_{=\supseteq} \overline{\mathcal{S}}$ and $\mathcal{S}' \supseteq_{=\supseteq} \mathcal{S}$ such that $\vdash_{\text{stack}} S' : \mathcal{S}' : \overline{\mathcal{S}}'$ and $\mathcal{S}' : \overline{\mathcal{S}}' \vdash_{\text{cval}} \mathsf{tt} : \mathsf{bool}$.

By Lemma 13, we conclude $\cdot; \cdot; \mathcal{S}' : \overline{\mathcal{S}}' \vdash_{\text{exp}} e_t : \tau, r'$.

Applying the induction hypothesis to (1a) $\vdash_{\text{stack}} S' : \mathcal{S}' : \overline{\mathcal{S}}'$, (1b) $\cdot; \cdot; \mathcal{S}' : \overline{\mathcal{S}}' \vdash_{\text{exp}} e_t : \tau, r'$, and (1c) $S'; e_t \hookrightarrow S''; v''$, we conclude that there exists $\overline{\mathcal{S}}'' \supseteq_{=\supseteq} \overline{\mathcal{S}}'$ and $\mathcal{S}'' \supseteq_{=\supseteq} \mathcal{S}'$ such that $\vdash_{\text{stack}} S'' : \mathcal{S}'' : \overline{\mathcal{S}}''$ and $\mathcal{S}'' : \overline{\mathcal{S}}' \vdash_{\text{cval}} v'' : \tau$.

By transitivity of $\supseteq_{=\supseteq}$, we conclude $\overline{\mathcal{S}}'' \supseteq_{=\supseteq} \overline{\mathcal{S}}$ and $\mathcal{S}'' \supseteq_{=\supseteq} \mathcal{S}$ such that $\vdash_{\text{stack}} S'' : \mathcal{S}'' : \overline{\mathcal{S}}''$ and $\mathcal{S}'' : \overline{\mathcal{S}}'' \vdash_{\text{cval}} v'' : \tau$.

Case $\dfrac{S; e_b \hookrightarrow S'; v' \qquad v' \equiv \mathsf{ff} \qquad S'; e_f \hookrightarrow S''; v''}{S; \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f \hookrightarrow S''; v''}$: ...

Case $\dfrac{\rho \equiv r \qquad r \in dom(S) \qquad l \notin dom(S(r))}{S; \lambda x : \tau.^{\theta'} e' \text{ at } \rho \hookrightarrow S\{(r,l) \mapsto \lambda x : \tau.^{\theta'} e'\}; \langle l \rangle_r}$: ... By inspection, the derivation of (1b) must end with

$$\dfrac{\begin{array}{c} \cdot; \cdot, x : \tau_1; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e' : \tau_2, \theta' \\ \cdot; \overline{\mathcal{S}} \vdash_{\text{place}} r \qquad \cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq r \end{array}}{\cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \lambda x : \tau_1.^{\theta'} e' \text{ at } r : (\tau_1 \xrightarrow{\theta'} \tau_2, r), r'}$$

Note that $\overline{\mathcal{S}}\{(r,l) \mapsto\} \supseteq_{=\supseteq} \overline{\mathcal{S}}$ and $\mathcal{S}\{(r,l) \mapsto \tau_1 \xrightarrow{\theta'} \tau_2\} \supseteq_{=\supseteq} \mathcal{S}$. Note that $\vdash_{\text{stack}} S\{(r,l) \mapsto \lambda x : \tau_1.^{\theta'} e'\} : \mathcal{S}\{(r,l) \mapsto \tau_1 \xrightarrow{\theta'} \tau_2\} : \overline{\mathcal{S}}\{(r,l) \mapsto\}$ (requires appealing to Lemma 13). Note that

$$\dfrac{\boxed{\vdash_{\text{stype}} \mathcal{S}\{(r,l) \mapsto \tau_1 \xrightarrow{\theta} \tau_2\} : \overline{\mathcal{S}}\{(r,l) \mapsto\}} \qquad \boxed{r \in dom(\overline{\mathcal{S}}\{(r,l) \mapsto\})} \qquad \boxed{l \in dom((\overline{\mathcal{S}}\{(r,l) \mapsto\})(r))} \qquad \boxed{\tau_1 \xrightarrow{\theta'} \tau_2 = (\mathcal{S}\{(r,l) \mapsto \tau_1 \xrightarrow{\theta'} \tau_2)(r,l)}}{\mathcal{S}\{(r,l) \mapsto \tau_1 \xrightarrow{\theta'} \tau_2\} : \overline{\mathcal{S}}\{(r,l) \mapsto\} \vdash_{\text{cval}} \langle l \rangle_r : (\tau_1 \xrightarrow{\theta'} \tau_2, r)}$$

Hence, $\overline{\mathcal{S}}\{(r,l) \mapsto\} \supseteq_{=\supseteq} \overline{\mathcal{S}}$, $\mathcal{S}\{(r,l) \mapsto \tau_1 \xrightarrow{\theta'} \tau_2\} \supseteq_{=\supseteq} \mathcal{S}$, $\vdash_{\text{stack}} S\{(r,l) \mapsto \lambda x : \tau_1.^{\theta'} e'\} : \mathcal{S}\{(r,l) \mapsto \tau_1 \xrightarrow{\theta'} \tau_2\} : \overline{\mathcal{S}}\{(r,l) \mapsto\}$, and $\mathcal{S}\{(r,l) \mapsto \tau_1 \xrightarrow{\theta'} \tau_2\} : \overline{\mathcal{S}}\{(r,l) \mapsto\} \vdash_{\text{cval}} \langle l \rangle_r : (\mathsf{int}, r)$, as required.

Case $\dfrac{\begin{array}{c} S; e_1 \hookrightarrow S_1; v_1 \qquad v_1 \equiv \langle l_1 \rangle_{r_1} \\ r_1 \in dom(S_1) \qquad l_1 \in dom(S_1(r_1)) \qquad S_1(r_1, l_1) = \lambda x : \tau_1.^{\theta'} e' \\ S_1; e_2 \hookrightarrow S_2; v_2 \qquad S_2; e'[v_2/x] \hookrightarrow S_3; v_3 \end{array}}{S; e_1\ e_2 \hookrightarrow S_3; v_3}$: By inspection, the derivation of (1b) must end with

$$\frac{\cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\tau_1 \xrightarrow{\theta'} \tau_2, \rho_1'), r' \qquad \cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq \rho_1' \qquad \cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_2 : \tau_1, r' \qquad \cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq \theta'}{\cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1\ e_2 : \tau_2, r'}$$

Applying the induction hypothesis to (1a) $\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}$, (1b) $\cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\tau_1 \xrightarrow{\theta'} \tau_2, \rho_1'), r'$, and (1c) $S; e_1 \hookrightarrow S_1; \langle l_1 \rangle_{r_1}$, we conclude that there exists $\overline{\mathcal{S}}_1 \supseteq_{=\supseteq} \overline{\mathcal{S}}$ and $\mathcal{S}_1 \supseteq_{=\supseteq} \mathcal{S}$ such that $\vdash_{\text{stack}} S_1 : \mathcal{S}_1 : \overline{\mathcal{S}}_1$ and $\mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{cval}} \langle l_1 \rangle_{r_1} : (\tau_1 \xrightarrow{\theta'} \tau_2, \rho_1')$.
By inspection, this derivation must end with

$$\frac{\vdash_{\text{stype}} \mathcal{S}_1 : \overline{\mathcal{S}}_1 \qquad r_1 \in dom(\overline{\mathcal{S}}_1) \qquad l \in dom(\overline{\mathcal{S}}_1(r_1)) \qquad \tau_1 \xrightarrow{\theta'} \tau_2 = \mathcal{S}_1(r_1, l_1)}{\mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{cval}} \langle l_1 \rangle_{r_1} : (\tau_1 \xrightarrow{\theta'} \tau_2, r_1)}$$

and $\rho_1' = r_1$.
Because $\vdash_{\text{stack}} S_1 : \mathcal{S}_1 : \overline{\mathcal{S}}_1$, we conclude $\mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{sto}} S_1(r_1, l_1) : \mathcal{S}_1(r_1, l_1)$. By inspection, this derivation must end with

$$\frac{\cdot; \cdot, x : \tau_1; \mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{exp}} e' : \tau_2, \theta'}{\mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{sto}} \lambda x : \tau_1.^{\theta'} e' : \tau_1 \xrightarrow{\theta'} \tau_2}$$

By Lemma 13, we conclude $\cdot; \cdot; \mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{exp}} e_2 : \tau_1, r'$.
Applying the induction hypothesis to (1a) $\vdash_{\text{stack}} S_1 : \mathcal{S}_1 : \overline{\mathcal{S}}_1$, (1b) $\cdot; \cdot; \mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{exp}} e_2 : \tau_1, r'$, and (1c) $S_1; e_2 \hookrightarrow S_2; v_2$, we conclude that there exists $\overline{\mathcal{S}}_2 \supseteq_{=\supseteq} \overline{\mathcal{S}}_1$ and $\mathcal{S}_2 \supseteq_{=\supseteq} \mathcal{S}_1$ such that $\vdash_{\text{stack}} S_2 : \mathcal{S}_2 : \overline{\mathcal{S}}_2$ and $\mathcal{S}_2 : \overline{\mathcal{S}}_2 \vdash_{\text{cval}} v_2 : \tau_1$.
By Lemma 13, we conclude $\cdot; \cdot, x : \tau_1; \mathcal{S}_2 : \overline{\mathcal{S}}_2 \vdash_{\text{exp}} e' : \tau_2, \theta'$.
By Lemma 11, we conclude $\cdot; \cdot; \mathcal{S}_2 : \overline{\mathcal{S}}_2 \vdash_{\text{exp}} e'[v_2/x] : \tau_2, \theta'$.
By Lemma 13, we conclude $\cdot; \overline{\mathcal{S}}_2 \vdash_{\text{rr}} r' \succeq \theta'$.
By Lemma 17 , we conclude $\theta' = r$. Hence, $\cdot; \cdot; \mathcal{S}_2 : \overline{\mathcal{S}}_2 \vdash_{\text{exp}} e'[v_2/x] : \tau_2, r$.
Applying the induction hypothesis to (1a) $\vdash_{\text{stack}} S_2 : \mathcal{S}_2 : \overline{\mathcal{S}}_2$, (1b) $\cdot; \cdot; \mathcal{S}_2 : \overline{\mathcal{S}}_2 \vdash_{\text{exp}} e'[v_2/x] : \tau_1, r$, and (1c) $S_2; e'[v_2/x] \hookrightarrow S_3; v_3$, we conclude that there exists $\overline{\mathcal{S}}_3 \supseteq_{=\supseteq} \overline{\mathcal{S}}_2$ and $\mathcal{S}_3 \supseteq_{=\supseteq} \mathcal{S}_2$ such that $\vdash_{\text{stack}} S_3 : \mathcal{S}_3 : \overline{\mathcal{S}}_3$ and $\mathcal{S}_3 : \overline{\mathcal{S}}_3 \vdash_{\text{cval}} v_3 : \tau_2$.
By transitivity of $\supseteq_{=\supseteq}$, we conclude $\overline{\mathcal{S}}_3 \supseteq_{=\supseteq} \overline{\mathcal{S}}$ and $\mathcal{S}_3 \supseteq_{=\supseteq} \mathcal{S}$ such that $\vdash_{\text{stack}} S_3 : \mathcal{S}_3 : \overline{\mathcal{S}}_3$ and $\mathcal{S}_3 : \overline{\mathcal{S}}_3 \vdash_{\text{cval}} v_3 : \tau_2$.

Case $\dfrac{\begin{array}{c} S; e_1 \hookrightarrow S_1; v_1 \\ S_1; e_2 \hookrightarrow S_2; v_2 \end{array} \quad \rho \equiv r \qquad r \in dom(S_2) \qquad l \notin dom(S_2(r))}{S; (e_1, e_2)\ \mathsf{at}\ \rho \hookrightarrow S_2\{(r,l) \mapsto (v_1, v_2)\}; \langle l \rangle_r}$: ...

Case $\dfrac{S; e \hookrightarrow S'; v \qquad v \equiv \langle l \rangle_r \qquad r \in dom(S') \qquad l \in dom(S'(r)) \qquad S'(r,l) = (v_1, v_2)}{S; \mathsf{fst}\ e \hookrightarrow S'; v_1}$: ...

Case $\dfrac{S; e \hookrightarrow S'; v \qquad v \equiv \langle l \rangle_r \qquad r \in dom(S') \qquad l \in dom(S'(r)) \qquad S'(r,l) = (v_1, v_2)}{S; \mathsf{snd}\ e \hookrightarrow S'; v_2}$: ...

Case $\dfrac{r \notin dom(S) \qquad S, r \mapsto \{\}; e[r/\varrho] \hookrightarrow S'; v'' \qquad S' \equiv S'', r \mapsto R''}{S; \mathsf{letregion}\ \varrho\ \mathsf{in}\ e \hookrightarrow S''[\bullet/r]; v''[\bullet/r]}$: By inspection, the derivation of (1b) must end with

$$\frac{\begin{array}{c} \cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau \\ \vdash_{\text{ctxt}} \cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}}; r' \\ \cdot, \varrho \succeq \{r'\}; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \varrho \end{array}}{\cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \mathsf{letregion}\ \varrho\ \mathsf{in}\ e : \tau, r'}$$

By Lemma 6, we conclude that $\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} r'$. By inspection of the derivation $\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} r'$, we conclude that $r' \in dom(\overline{\mathcal{S}})$. Hence $\overline{\mathcal{S}} = \overline{\mathcal{S}}_1, r' \mapsto \overline{\mathcal{R}}', \overline{\mathcal{S}}_2$.

By inspection of the derivation $\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}$, we conclude $\vdash_{\text{sdom}} \overline{\mathcal{S}}$, $\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}$, and $dom(S) = dom(\mathcal{S}) = dom(\overline{\mathcal{S}})$. Note that $\overline{\mathcal{S}}, r \mapsto \{\} \supseteq_{\supseteq=} \overline{\mathcal{S}}$ and $\mathcal{S}, r \mapsto \{\} \supseteq_{\supseteq=} \mathcal{S}$ and

$$\frac{\boxed{\vdash_{\text{sdom}} \overline{\mathcal{S}}} \qquad \boxed{r \notin dom(\overline{\mathcal{S}})} \qquad \boxed{\overline{\vdash_{\text{rdom}} \{\}}}}{\vdash_{\text{sdom}} \overline{\mathcal{S}}, r \mapsto \{\}}$$

and $\vdash_{\text{stype}} \mathcal{S}, r \mapsto \{\} : \overline{\mathcal{S}}, r \mapsto \{\}$ (requires appealing to Lemma 13) and $\vdash_{\text{stack}} S, r \mapsto \{\} : \mathcal{S}, r \mapsto \{\} : \overline{\mathcal{S}}, r \mapsto \{\}$ (requires appealing to Lemma 13).

Note that

$$\frac{\dfrac{\boxed{\vdash_{\text{sdom}} \overline{\mathcal{S}}, r \mapsto \{\}}}{\overline{\mathcal{S}}, r \mapsto \{\} \vdash_{\text{rctxt}} \cdot}}{\cdot; \overline{\mathcal{S}}, r \mapsto \{\} \vdash_{\text{re}} r \succeq \{r'\}} \qquad \frac{\dfrac{\boxed{\vdash_{\text{sdom}} \overline{\mathcal{S}}, r \mapsto \{\}}}{\overline{\mathcal{S}}, r \mapsto \{\} \vdash_{\text{rctxt}} \cdot} \qquad \overline{\mathcal{S}}, r \mapsto \{\} = \overline{\mathcal{S}}_1, r' \mapsto \overline{\mathcal{R}}', \overline{\mathcal{S}}_2, r \mapsto \{\}}{\cdot; \overline{\mathcal{S}}, r \mapsto \{\} \vdash_{\text{rr}} r \succeq r'}$$

By Lemma 13, we conclude $\cdot, \varrho \succeq \{r'\}; \cdot; \mathcal{S}, r \mapsto \{\} : \overline{\mathcal{S}}, r \mapsto \{\} \vdash_{\text{exp}} e : \tau, \varrho$, By Lemma 10 we conclude $\cdot; \cdot; \mathcal{S}, r \mapsto \{\} : \overline{\mathcal{S}}, r \mapsto \{\} \vdash_{\text{exp}} e[r/\varrho] : \tau[r/\varrho], r$. By Lemma 15, we conclude $\tau[r/\varrho] = \tau$. Hence, $\cdot; \cdot; \mathcal{S}, r \mapsto \{\} : \overline{\mathcal{S}}, r \mapsto \{\} \vdash_{\text{exp}} e[r/\varrho] : \tau, r$.

Applying the induction hypothesis to (1a) $\vdash_{\text{stack}} S, r \mapsto \{\} : \mathcal{S}, r \mapsto \{\} : \overline{\mathcal{S}}, r \mapsto \{\}$, (1b) $\cdot; \cdot; \mathcal{S}, r \mapsto \{\} : \overline{\mathcal{S}}, r \mapsto \{\} \vdash_{\text{exp}} e[r/\varrho] : \tau, r$, (1c) $S, r \mapsto \{\}; e[r/\varrho] \hookrightarrow S'', r \mapsto R''; v''$, we conclude that there exists $\overline{\mathcal{S}}^\dagger \supseteq_{=\supseteq} \overline{\mathcal{S}}, r \mapsto \{\}$ and $\mathcal{S}^\dagger \supseteq_{=\supseteq} \mathcal{S}, r \mapsto \{\}$ such that $\vdash_{\text{stack}} S'', r \mapsto R'' : \mathcal{S}^\dagger : \overline{\mathcal{S}}^\dagger$ and $\mathcal{S}^\dagger : \overline{\mathcal{S}}^\dagger \vdash_{\text{cval}} v'' : \tau$. Note that $\overline{\mathcal{S}}^\dagger \supseteq_{=\supseteq} \overline{\mathcal{S}}, r \mapsto \{\}$ implies $\overline{\mathcal{S}}^\dagger \equiv \overline{\mathcal{S}}'', r \mapsto \overline{\mathcal{R}}''$. Note that $\mathcal{S}^\dagger \supseteq_{=\supseteq} \mathcal{S}, r \mapsto \{\}$ implies $\mathcal{S}^\dagger \equiv \mathcal{S}'', r \mapsto \mathcal{R}''$. Hence, we conclude that there exists $\overline{\mathcal{S}}'' \supseteq_{=\supseteq} \overline{\mathcal{S}}$ and $\overline{\mathcal{R}}'' \supseteq_{\supseteq} \{\}$ and $\mathcal{S}'' \supseteq_{=\supseteq} \mathcal{S}$ and $\mathcal{R}'' \supseteq_{\supseteq} \{\}$ such that $\mathcal{S}'', r \mapsto \mathcal{R}'' : \overline{\mathcal{S}}'', r \mapsto \overline{\mathcal{R}}'' \vdash_{\text{cval}} v'' : \tau$. By Lemma 14, we conclude $\vdash_{\text{stack}} S''[\bullet/r] : \mathcal{S}''[\bullet/r] : \overline{\mathcal{S}}''[\bullet/r]$. By Lemma 14, we conclude $\mathcal{S}''[\bullet/r] : \overline{\mathcal{S}}''[\bullet/r] \vdash_{\text{cval}} v''[\bullet/r] : \tau[\bullet/r]$. By the derivation (1b), we conclude $\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau$. By Lemma 16, we conclude $\tau[\bullet/r] = \tau$. Note that $\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}$ implies $\forall r \in dom(\mathcal{S}). \forall l \in dom(\mathcal{S}(r)). \cdot; \mathcal{S} \vdash_{\text{btype}} \mathcal{S}(r, l)$. Note that $\mathcal{S}'' \supseteq_{=\supseteq} \mathcal{S}$ implies $\forall r \in dom(\mathcal{S}). \forall l \in dom(\mathcal{S}(r)). \mathcal{S}''(r, l) = \mathcal{S}(r, l)$. Hence, $\forall r \in dom(\mathcal{S}). \forall l \in dom(\mathcal{S}(r)). \cdot; \overline{\mathcal{S}} \vdash_{\text{btype}} \mathcal{S}''(r, l)$. By Lemma 16 (using $r \notin dom(\mathcal{S})$), we conclude $\forall r \in dom(\mathcal{S}). \forall l \in dom(\mathcal{S}(r)). \mathcal{S}''(r, l)[\bullet/r] \equiv \mathcal{S}''(r, l)$. Furthermore, $\forall r \in dom(\mathcal{S}). \forall l \in dom(\mathcal{S}(r)). \mathcal{S}''[\bullet/r](r, l) \equiv \mathcal{S}''(r, l)$. Hence, $\forall r \in dom(\mathcal{S}). \forall l \in dom(\mathcal{S}(r)). \mathcal{S}''[\bullet/r](r, l) = \mathcal{S}(r, l)$. Hence, $\mathcal{S}''[\bullet/r] \supseteq_{=\supseteq} \mathcal{S}$. Hence, $\overline{\mathcal{S}}'' \supseteq_{=\supseteq} \overline{\mathcal{S}}$, $\mathcal{S}''[\bullet/r] \supseteq_{=\supseteq} \mathcal{S}$, $\vdash_{\text{stack}} S''[\bullet/r] : \mathcal{S}''[\bullet/r] : \overline{\mathcal{S}}''$, and $\mathcal{S}''[\bullet/r] : \overline{\mathcal{S}}'' \vdash_{\text{cval}} v''[\bullet/r] : \tau$, as required.

Case $\dfrac{\rho \equiv r \qquad r \in dom(S) \qquad l \notin dom(S(r))}{S; \lambda\varrho \succeq \varphi.^{\theta'} u' \text{ at } \rho \hookrightarrow S\{(r, l) \mapsto \lambda\varrho \succeq \varphi.^{\theta'} u'\}; \langle l \rangle_r} : \ldots$

Case $\dfrac{\begin{array}{c} S; e_1 \hookrightarrow S_1; v_1 \qquad v_1 = \langle l_1 \rangle_{r_1} \\ r_1 \in dom(S_1) \qquad l_1 \in dom(S_1(r_1)) \qquad S_1(r_1, l_1) = \lambda\varrho \succeq \varphi.^{\theta'} u' \\ S_1; u'[\rho_2/\varrho] \hookrightarrow S_2; v_2 \end{array}}{S; e_1\ [\rho_2] \hookrightarrow S_2; v_2}$: By inspection, the derivation of (1b) must end with

$$\frac{\begin{array}{c} \cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\Pi\varrho \succeq \varphi.^{\theta'} \tau, \rho_1'), r' \qquad \cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq \rho_1' \\ \cdot; \overline{\mathcal{S}} \vdash_{\text{place}} \rho_2 \qquad \cdot; \overline{\mathcal{S}} \vdash_{\text{re}} \rho_2 \succeq \varphi \\ \cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq \theta'[\rho_2/\varrho] \end{array}}{\cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1\ [\rho_2] : \tau[\rho_2/\varrho], r'}$$

Applying the induction hypothesis to (1a) $\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}$, (1b) $\cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\Pi\varrho \succeq \varphi.^{\theta'} \tau, \rho_1'), r'$, and (1c) $S; e_1 \hookrightarrow S_1; \langle l_1 \rangle_{r_1}$, we conclude that there exists $\overline{\mathcal{S}}_1 \supseteq_{=\supseteq} \overline{\mathcal{S}}$ and $\mathcal{S}_1 \supseteq_{=\supseteq} \mathcal{S}$ such that $\vdash_{\text{stack}} S_1 : \mathcal{S}_1 : \overline{\mathcal{S}}_1$ and $\mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{cval}} \langle l_1 \rangle_{r_1} : (\Pi\varrho \succeq \varphi.^{\theta'} \tau, \rho_1')$.

By inspection, this derivation must end with

$$\frac{\vdash_{\text{stype}} \mathcal{S}_1 : \overline{\mathcal{S}}_1 \qquad r_1 \in dom(\overline{\mathcal{S}}_1) \qquad l_1 \in dom(\overline{\mathcal{S}}_1(r_1)) \qquad \Pi\varrho \succeq \varphi.^{\theta'}\tau = \mathcal{S}(r_1, l_1)}{\mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{cval}} \langle l_1 \rangle_{r_1} : (\Pi\varrho \succeq \varphi.^{\theta'}\tau, r_1).}$$

and $\rho_1' = r_1$.

Because $\vdash_{\text{stack}} S_1 : \mathcal{S}_1 : \overline{\mathcal{S}}_1$, we conclude $\mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{sto}} S_1(r_1, l_1) : \mathcal{S}_1(r_1, l_1)$. By inspection, this derivation must end with

$$\frac{\cdot, \varrho \succeq \varphi; \cdot; \mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{exp}} u' : \tau, \theta'}{\mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{sto}} \lambda\varrho \succeq \varphi.^{\theta'} u' : \Pi\varrho \succeq \varphi.^{\theta'}\tau}$$

By Lemma 13, we conclude $\cdot; \overline{\mathcal{S}}_1 \vdash_{\text{re}} \rho_2 \succeq \varphi$. By Lemma 10 we conclude $\cdot; \cdot; \mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{exp}} u'[\rho_2/\varrho] : \tau[\rho_2/\varrho], \theta'[\rho_2/\varrho]$. By Lemma 13, we conclude $\cdot; \overline{\mathcal{S}}_1 \vdash_{\text{rr}} r' \succeq \theta'[\rho_2/\varrho]$. By Lemma 17, we conclude $\theta'[\rho_2/\varrho] = r$. Hence $\cdot; \cdot; \mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{exp}} u'[\rho_2/\varrho] : \tau[\rho_2/\varrho], r$.

Applying the induction hypothesis to (1a) $\vdash_{\text{stack}} S_1 : \mathcal{S}_1 : \overline{\mathcal{S}}_1$, (1b) $\cdot; \cdot; \mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{exp}} u'[\rho_2/\varrho] : \tau[\rho_2/\varrho], r$, and (1c) $S_1; u'[\rho_2/\varrho] \hookrightarrow S_2; v_2$, we conclude that there exists $\overline{\mathcal{S}}_2 \supseteq_{=\supseteq} \overline{\mathcal{S}}_1$ and $\mathcal{S}_2 \supseteq_{=\supseteq} \mathcal{S}_1$ such that $\vdash_{\text{stack}} S_2 : \mathcal{S}_2 : \overline{\mathcal{S}}_2$ and $\mathcal{S}_2 : \overline{\mathcal{S}}_2 \vdash_{\text{cval}} v_2 : \tau[\rho_2/\varrho]$.

By transitivity of $\supseteq_{=\supseteq}$, we conclude $\overline{\mathcal{S}}_2 \supseteq_{=\supseteq} \overline{\mathcal{S}}$ and $\mathcal{S}_2 \supseteq_{=\supseteq} \mathcal{S}$ such that $\vdash_{\text{stack}} S_2 : \mathcal{S}_2 : \overline{\mathcal{S}}_2$ and $\mathcal{S}_2 : \overline{\mathcal{S}}_2 \vdash_{\text{cval}} v_2 : \tau[\rho_2/\varrho]$.

□

2.8 The Bounded Region Calculus

While the Single Effect Calculus has many similarities to other region calculi, it is not immediately clear that the "single effect" restriction does not inhibit the ability of SEC to model realistic region calculi. In this section, we present a core model of Cyclone, called the Bounded Region Calculus (BRC); alternatively, one can view the Single Effect Calculus as a restricted form of the Bounded Region Calculus.

One key difference (among many) between Cyclone and the Tofte-Talpin region calculus is that the type-and-effects system of Cyclone extends that of Tofte-Talpin's with the form of region subtyping introduced in the Single Effect Calculus. However, like the Tofte-Talpin region calculus, Cyclone treats effects as sets of regions affected by the evaluation of an expression. The Bounded Region Calculus combines these traits by admitting both latent effects as sets of regions *and* bounded region polymorphism.

2.8.1 Surface Syntax of BRC

Figure 12 presents the syntax of "surface programs" (that is, excluding intermediate terms that appear in the operational semantics) of the Bounded Effect Calculus.

As in the Single Effect Calculus, we distinguish between places and effects. The Bounded Region Calculus includes the same class of effects – essentially, that of a finite set of places. We also distinguish between types and boxed types; note that the "single effect" θ in function and region abstraction boxed types of the Single Effect Calculus are replaced by a "general effect" φ in the Bounded Region Calculus. Also note that region abstraction types continue to include a region bound.

The Bounded Region Calculus includes all the terms seen previously in the Single Effect Calculus. Once again, note that the "single effect" θ in function and region abstractions are replaced by a general effect φ in the Tofte-Talpin Calculus. Also note that region abstractions continue to include a region bound.

2.8.2 Static Semantics of BRC

Figures 13, 14, 15, 16, 17, and 18 define the static semantics of the Bounded Effect Calculus Calculus. The development is entirely analagous to that of the Single Effect Calculus.

$$
\begin{array}{rrcll}
 & i & \in & \mathbb{Z} & \\
 & \varepsilon & \in & EVars^{BRC} & \\
 & \vartheta, \varrho & \in & RVars^{BRC} & \text{where } \mathcal{H} \in RVars^{BRC} \\
 & f, x & \in & Vars^{BRC} & \\
 & & & & \\
\text{Surface region placeholders} & \theta, \rho & ::= & \varrho & \\
\text{Surface effects} & \varphi & ::= & \{\rho_1, \ldots, \rho_n\} & \\
\text{Surface types} & \tau & ::= & \mathsf{bool} \mid (\mu, \rho) & \\
\text{Surface boxed types} & \mu & ::= & \mathsf{int} \mid \tau_1 \xrightarrow{\varphi} \tau_2 \mid \tau_1 \times \tau_2 \mid \Pi \varrho \succeq \varphi'.^{\varphi} \tau & \\
 & & & & \\
\text{Surface programs} & p & ::= & e & \\
\text{Surface terms} & e & ::= & i \ \mathsf{at}\ \rho \mid e_1 \oplus e_2 \ \mathsf{at}\ \rho \mid e_1 \otimes e_2 \mid & \\
 & & & \mathsf{tt} \mid \mathsf{ff} \mid \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f \mid & \\
 & & & x \mid \lambda x : \tau.^{\varphi} e \ \mathsf{at}\ \rho \mid e_1\ e_2 \mid & \\
 & & & (e_1, e_2) \ \mathsf{at}\ \rho \mid \mathsf{fst}\ e \mid \mathsf{snd}\ e \mid & \\
 & & & \mathsf{letregion}\ \varrho.e \mid \lambda \varrho \succeq \varphi'.^{\varphi} u \ \mathsf{at}\ \rho \mid e\ [\rho] & \\
\text{Surface abstractions} & u & ::= & \lambda x : \tau.^{\varphi} e \ \mathsf{at}\ \rho \mid \lambda \varrho \succeq \varphi'.^{\varphi} u \ \mathsf{at}\ \rho & \\
\text{Surface values} & v & ::= & \mathsf{tt} \mid \mathsf{ff} \mid x &
\end{array}
$$

Figure 12: Surface syntax of BRC

$$
\begin{array}{rrcl}
\text{Region contexts} & \Delta & ::= & \cdot \mid \Delta, \varrho \succeq \varphi \\
\text{Value contexts} & \Gamma & ::= & \cdot \mid \Gamma, x : \tau
\end{array}
$$

Figure 13: Static semantics of BRC (definitions)

We summarize the main typing judgements in the following table:

Judgement	Meaning
$\Delta \vdash_{\text{btype}} \mu$	Boxed type μ is well-formed.
$\Delta \vdash_{\text{type}} \tau$	Type τ is well-formed.
$\Delta \vdash_{\text{rr}} \rho \succeq \rho'$	If region ρ is live, then region ρ' is live. (Alt.: region ρ' outlives region ρ.)
$\Delta \vdash_{\text{re}} \rho \succeq \varphi$	If region ρ is live, then all regions in φ are live. (Alt.: all regions in φ outlive region ρ.)
$\Delta \vdash_{\text{er}} \varphi \ni \rho$	Region ρ is a region in φ.
$\Delta \vdash_{\text{ee}} \varphi \supseteq \varphi'$	All region in φ' are regions in φ.
$\Delta; \Gamma \vdash_{\text{exp}} e : \tau, \varphi$	Term e has type τ and effect φ.
$\vdash_{\text{prog}} p \ \mathsf{ok}$	Program p is well-typed.

2.8.3 Translation of BRC to SEC

We can give a straightforward translation from the Bounded EffectCalculus into the Single Effect Calculus.

At the type level, this transation expands every function type into a region abstraction and function type:

$$\mathbb{T}^{\tau}_{\tau}\left[\!\left[(\tau_1 \xrightarrow{\varphi} \tau_2, \rho)\right]\!\right] = (\Pi\vartheta \succeq \varphi.^{\rho}(\mathbb{T}^{\tau}_{\tau}\left[\!\left[\tau_1\right]\!\right] \xrightarrow{\vartheta} \mathbb{T}^{\tau}_{\tau}\left[\!\left[\tau_2\right]\!\right], \rho), \rho)$$

At the term level, source functions become region abstractions and functions and applications become region instantiations and applications. A similar approach deals with region abstractions in the source language.

$\boxed{\Delta;\Gamma \vdash_{\text{exp}} e : \tau, \varphi}$

$$\frac{\vdash_{\text{ctxt}} \Delta;\Gamma;\varphi \qquad \Delta \vdash_{\text{place}} \rho \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho}{\Delta;\Gamma \vdash_{\text{exp}} i \text{ at } \rho : (\mathsf{int}, \rho), \varphi}$$

$$\frac{\Delta;\Gamma \vdash_{\text{exp}} e_1 : (\mathsf{int}, \rho_1), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho_1 \qquad \Delta;\Gamma \vdash_{\text{exp}} e_2 : (\mathsf{int}, \rho_2), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho_2 \qquad \Delta \vdash_{\text{place}} \rho \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho}{\Delta;\Gamma \vdash_{\text{exp}} e_1 \oplus e_2 \text{ at } \rho : (\mathsf{int}, \rho), \varphi}$$

$$\frac{\Delta;\Gamma \vdash_{\text{exp}} e_1 : (\mathsf{int}, \rho_1), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho_1 \qquad \Delta;\Gamma \vdash_{\text{exp}} e_2 : (\mathsf{int}, \rho_2), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho_2}{\Delta;\Gamma \vdash_{\text{exp}} e_1 \otimes e_2 : \mathsf{bool}, \varphi}$$

$$\frac{\vdash_{\text{ctxt}} \Delta;\Gamma;\varphi}{\Delta;\Gamma \vdash_{\text{exp}} \mathsf{tt} : \mathsf{bool}, \varphi} \qquad \frac{\vdash_{\text{ctxt}} \Delta;\Gamma;\varphi}{\Delta;\Gamma \vdash_{\text{exp}} \mathsf{ff} : \mathsf{bool}, \varphi}$$

$$\frac{\Delta;\Gamma \vdash_{\text{exp}} e_b : \mathsf{bool}, \varphi \qquad \Delta;\Gamma \vdash_{\text{exp}} e_t : \tau, \varphi \qquad \Delta;\Gamma \vdash_{\text{exp}} e_f : \tau, \varphi}{\Delta;\Gamma \vdash_{\text{exp}} \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f : \tau, \varphi} \qquad \frac{\vdash_{\text{ctxt}} \Delta;\Gamma;\varphi \qquad x \in dom(\Gamma) \qquad \Gamma(x) = \tau}{\Delta;\Gamma \vdash_{\text{exp}} x : \tau, \varphi}$$

$$\frac{\Delta;\Gamma, x : \tau_1 \vdash_{\text{exp}} e' : \tau_2, \varphi' \qquad \Delta \vdash_{\text{place}} \rho \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho}{\Delta;\Gamma \vdash_{\text{exp}} \lambda x : \tau_1.^{\varphi'} e' \text{ at } \rho : (\tau_1 \xrightarrow{\varphi'} \tau_2, \rho), \varphi}$$

$$\frac{\Delta;\Gamma \vdash_{\text{exp}} e_1 : (\tau_1 \xrightarrow{\varphi'} \tau_2, \rho_1'), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho_1' \qquad \Delta;\Gamma \vdash_{\text{exp}} e_2 : \tau_1, \varphi \qquad \Delta \vdash_{\text{ee}} \varphi \supseteq \varphi'}{\Delta;\Gamma \vdash_{\text{exp}} e_1\ e_2 : \tau_2, \varphi}$$

$$\frac{\Delta;\Gamma \vdash_{\text{exp}} e_1 : \tau_1, \varphi \qquad \Delta;\Gamma \vdash_{\text{exp}} e_2 : \tau_2, \varphi \qquad \Delta \vdash_{\text{place}} \rho \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho}{\Delta;\Gamma \vdash_{\text{exp}} (e_1, e_2) \text{ at } \rho : (\tau_1 \times \tau_2, \rho), \varphi}$$

$$\frac{\Delta;\Gamma \vdash_{\text{exp}} e : (\tau_1 \times \tau_2, \rho), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho}{\Delta;\Gamma \vdash_{\text{exp}} \mathsf{fst}\ \ e : \tau_1, \varphi}$$

$$\frac{\Delta;\Gamma \vdash_{\text{exp}} e : (\tau_1 \times \tau_2, \rho), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho}{\Delta;\Gamma \vdash_{\text{exp}} \mathsf{snd}\ \ e : \tau_2, \varphi}$$

$$\frac{\Delta \vdash_{\text{type}} \tau \qquad \vdash_{\text{ctxt}} \Delta;\Gamma;\{\rho_1, \ldots, \rho_n\} \qquad \Delta, \varrho \succeq \{\rho_1, \ldots, \rho_n\};\Gamma \vdash_{\text{exp}} e : \tau, \{\rho_1, \ldots, \rho_n, \varrho\}}{\Delta;\Gamma \vdash_{\text{exp}} \mathsf{letregion}\ \varrho.e : \tau, \{\rho_1, \ldots, \rho_n\}}$$

$$\frac{\Delta, \varrho \succeq \varphi'';\Gamma \vdash_{\text{exp}} u' : \tau, \varphi' \qquad \Delta \vdash_{\text{place}} \rho \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho}{\Delta;\Gamma \vdash_{\text{exp}} \lambda\varrho \succeq \varphi''.^{\varphi'} u' \text{ at } \rho : (\Pi\varrho.^{\varphi'} \tau, \rho), \varphi}$$

$$\frac{\Delta;\Gamma \vdash_{\text{exp}} e : (\Pi\varrho \succeq \varphi''.^{\varphi'} \tau, \rho_1'), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho_1' \qquad \Delta \vdash_{\text{place}} \rho_2 \qquad \Delta \vdash_{\text{re}} \rho_2 \succeq \varphi'' \qquad \Delta \vdash_{\text{ee}} \varphi \supseteq \varphi'[\rho_2/\varrho]}{\Delta;\Gamma \vdash_{\text{exp}} e\ [\rho_2] : \tau[\rho_2/\varrho], \varphi}$$

$$\frac{\Delta;\Gamma, f : \tau \vdash_{\text{exp}} u : \tau, \varphi}{\Delta;\Gamma \vdash_{\text{exp}} \mathsf{fix}\ f : \tau.u : \tau, \varphi}$$

Figure 14: Static semantics of BRC (expressions)

$\boxed{\Delta \vdash_{\text{rr}} \rho \succeq \rho'}$

$$\frac{\vdash_{\text{rctxt}} \Delta \quad \Delta(\varrho) = \{\rho_1, \ldots, \rho_i, \ldots, \rho_n\}}{\Delta \vdash_{\text{rr}} \varrho \succeq \rho_i} \qquad \frac{\Delta \vdash_{\text{place}} \rho}{\Delta \vdash_{\text{rr}} \rho \succeq \rho} \qquad \frac{\Delta \vdash_{\text{rr}} \rho \succeq \rho' \quad \Delta \vdash_{\text{rr}} \rho' \succeq \rho''}{\Delta \vdash_{\text{rr}} \rho \succeq \rho''}$$

$\boxed{\Delta \vdash_{\text{re}} \rho \succeq \varphi}$

$$\frac{\vdash_{\text{rctxt}} \Delta \quad \Delta \vdash_{\text{rr}} \rho \succeq \rho_i \;^{i \in 1 \ldots n}}{\Delta \vdash_{\text{re}} \rho \succeq \{\rho_1, \ldots, \rho_n\}}$$

$\boxed{\Delta \vdash_{\text{er}} \varphi \ni \rho}$

$$\frac{\Delta \vdash_{\text{eff}} \{\rho_1, \ldots, \rho_n\}}{\Delta \vdash_{\text{er}} \{\rho_1, \ldots, \rho_n\} \ni \rho_i}$$

$\boxed{\Delta \vdash_{\text{re}} \varphi \supseteq \varphi'}$

$$\frac{\Delta \vdash_{\text{eff}} \varphi \quad \Delta \vdash_{\text{er}} \varphi \ni \rho_i \;^{i \in 1 \ldots n}}{\Delta \vdash_{\text{ee}} \varphi \supseteq \{\rho_1, \ldots, \rho_n\}}$$

Figure 15: Static semantics of BRC (casts)

$\boxed{\Delta \vdash_{\text{place}} \rho}$

$$\frac{\vdash_{\text{rctxt}} \Delta \quad \varrho \in dom(\Delta)}{\Delta \vdash_{\text{place}} \varrho}$$

$\boxed{\Delta \vdash_{\text{eff}} \varphi}$

$$\frac{\vdash_{\text{rctxt}} \Delta \quad \Delta \vdash_{\text{place}} \rho_i \;^{i \in 1 \ldots n}}{\Delta \vdash_{\text{eff}} \{\rho_1, \ldots, \rho_n\}}$$

$\boxed{\Delta \vdash_{\text{btype}} \mu}$

$$\frac{\vdash_{\text{rctxt}} \Delta}{\Delta \vdash_{\text{btype}} \mathsf{int}} \qquad \frac{\Delta \vdash_{\text{type}} \tau_1 \quad \Delta \vdash_{\text{eff}} \varphi \quad \Delta \vdash_{\text{type}} \tau_2}{\Delta \vdash_{\text{btype}} \tau_1 \xrightarrow{\varphi} \tau_2} \qquad \frac{\Delta \vdash_{\text{type}} \tau_1 \quad \Delta \vdash_{\text{type}} \tau_2}{\Delta \vdash_{\text{btype}} \tau_1 \times \tau_2}$$

$$\frac{\Delta \vdash_{\text{eff}} \varphi' \quad \Delta, \varrho \succeq \varphi' \vdash_{\text{eff}} \varphi \quad \Delta, \varrho \succeq \varphi' \vdash_{\text{type}} \tau}{\Delta \vdash_{\text{btype}} \Pi \varrho \succeq \varphi' .^{\varphi} \tau}$$

$\boxed{\Delta \vdash_{\text{type}} \tau}$

$$\frac{\vdash_{\text{rctxt}} \Delta}{\Delta \vdash_{\text{type}} \mathsf{bool}} \qquad \frac{\Delta \vdash_{\text{btype}} \mu \quad \Delta \vdash_{\text{place}} \rho}{\Delta \vdash_{\text{type}} (\mu, \rho)}$$

Figure 16: Static semantics of BRC (types)

$\boxed{\vdash_{\text{rctxt}} \Delta}$

$$\frac{}{\vdash_{\text{rctxt}} \cdot} \qquad \frac{\vdash_{\text{rctxt}} \Delta \qquad \varrho \notin dom(\Delta) \qquad \Delta \vdash_{\text{eff}} \varphi}{\vdash_{\text{rctxt}} \Delta, \varrho \succeq \varphi}$$

$\boxed{\Delta \vdash_{\text{vctxt}} \Gamma}$

$$\frac{\vdash_{\text{rctxt}} \Delta}{\Delta \vdash_{\text{vctxt}} \cdot} \qquad \frac{\Delta \vdash_{\text{vctxt}} \Gamma \qquad x \notin dom(\Gamma) \qquad \Delta \vdash_{\text{type}} \tau}{\Delta \vdash_{\text{vctxt}} \Gamma, x : \tau}$$

$\boxed{\vdash_{\text{ctxt}} \Delta; \Gamma; \varphi}$

$$\frac{\Delta \vdash_{\text{vctxt}} \Gamma \qquad \Delta \vdash_{\text{eff}} \varphi}{\vdash_{\text{ctxt}} \Delta; \Gamma; \varphi}$$

Figure 17: Static semantics of BRC (contexts)

$\boxed{\vdash_{\text{prog}} p \ \mathsf{ok}}$

$$\frac{\cdot, \mathcal{H} \succeq \{\}; \cdot \vdash_{\text{exp}} p : \mathsf{bool}, \{\mathcal{H}\}}{\vdash_{\text{prog}} p \ \mathsf{ok}}$$

Figure 18: Static semantics of BRC (surface programs)

Essentially, this translation works by looking for the places where region sets are used in the source calculus and simply replacing them by an abstraction over that set. Clearly, this is not the most efficient translation. For example, in places where we could statically identify an upper bound on the region set (e.g., a singleton region set), we could elide the abstraction and simply use the upper bound.

We start with a few preliminaries. We assume injections from the sets $RVars^{BRC}$ and $Vars^{BRC}$ to the sets $RVars^{SEC}$ and $Vars^{SEC}$ respectively. In the translation, applications of such injections will be clear from context and we freely use variables from source objects in target objects. We further assume a partitioning

$$RVars^{SEC} = RVars^{BRC} \uplus \Theta$$

and draw ϑ region variables from the set Θ. Hence, no ϑ region variable appears in any source BRC program.

Figure 19 gives the translation from the Bounded Effect Calculus to the Single Effect Calculus. We prove that the translation is type-preserving. (Note that we adopt the simplified type-system for the Single Effect Calculus (see Section 2.6.5), where we assume empty stack types and stack domains.)

Lemma 19 (Translation preserves types)

(1) If $\vdash_{\text{rctxt}} \Delta$, *then* $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!]$.
Furthermore, $dom(\Delta) = dom(\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!])$.

(2) If $\Delta \vdash_{\text{place}} \rho$, *then* $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!] \vdash_{\text{place}} \rho$.

(3) If $\Delta \vdash_{\text{eff}} \varphi$, *then* $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!] \vdash_{\text{eff}} \varphi$.

(4) If $\Delta \vdash_{\text{btype}} \mu$, *then forall* Δ' *and* ρ,
if $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{place}} \rho$, *then* $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{btype}} \mathbb{T}^{\mu}_{\mu}[\![\mu]\!]_{\rho}$.

(5) If $\Delta \vdash_{\text{type}} \tau$, *then forall* Δ',
if $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'$, *then* $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{type}} \mathbb{T}^{\tau}_{\tau}[\![\tau]\!]$.

(6) If $\Delta \vdash_{\text{vctxt}} \Gamma$, *then* $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!] \vdash_{\text{vctxt}} \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!]$.

(7) If $\Delta; \Gamma \vdash_{\text{exp}} e : \tau, \varphi$, *then forall* Δ' *and* θ,
if $\vdash_{\text{ctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!]; \theta$ *and* $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{re}} \theta \succeq \varphi$,
then $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \mathbb{T}^{e}_{e}[\![e]\!]_{\theta} : \mathbb{T}^{\tau}_{\tau}[\![\tau]\!], \theta$.

(8) If $\vdash_{\text{prog}} p$ ok,
then $\vdash_{\text{prog}} \mathbb{T}^{p}_{p}[\![p]\!]$ ok.

Proof.

(1,2,3) By induction on the derivations $\vdash_{\text{rctxt}} \Delta$, $\Delta \vdash_{\text{place}} \rho$, and $\Delta \vdash_{\text{eff}} \varphi$.

(4,5) Proceed by induction on the derivations $\Delta \vdash_{\text{btype}} \mu$ and $\Delta \vdash_{\text{type}} \mu$. Note that by applying Lemma 2 to $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{place}} \rho$, we conclude that $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'$.

Case $\dfrac{\vdash_{\text{rctxt}} \Delta}{\Delta \vdash_{\text{btype}} \mathsf{int}}$: We are required to show

$$\frac{\boxed{\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \ ^{\checkmark}}}{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{btype}} \mathsf{int}}$$

Case $\dfrac{\Delta \vdash_{\text{type}} \tau_1 \qquad \Delta \vdash_{\text{eff}} \varphi \qquad \Delta \vdash_{\text{type}} \tau_2}{\Delta \vdash_{\text{btype}} \tau_1 \xrightarrow{\varphi} \tau_2}$: We are required to show

$$\frac{\boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{eff}} \varphi} \qquad \boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi \vdash_{\text{place}} \rho}}{\dfrac{\boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi \vdash_{\text{type}} \mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!]} \qquad \boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi \vdash_{\text{place}} \vartheta} \qquad \boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi \vdash_{\text{type}} \mathbb{T}^{\tau}_{\tau}[\![\tau_2]\!]}}{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi \vdash_{\text{type}} (\mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!] \xrightarrow{\vartheta} \mathbb{T}^{\tau}_{\tau}[\![\tau_2]\!], \rho)}}{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{btype}} \Pi\vartheta \succeq \varphi.^{\rho}(\mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!] \xrightarrow{\vartheta} \mathbb{T}^{\tau}_{\tau}[\![\tau_2]\!], \rho)}$$

Region contexts

$$\begin{aligned}
\mathbb{T}^{\Delta}_{\Delta}\left[\!\left[\cdot\right]\!\right] &= \cdot \\
\mathbb{T}^{\Delta}_{\Delta}\left[\!\left[\Delta, \varrho \succeq \varphi\right]\!\right] &= \mathbb{T}^{\Delta}_{\Delta}\left[\!\left[\Delta\right]\!\right], \varrho \succeq \varphi
\end{aligned}$$

Boxed types

$$\begin{aligned}
\mathbb{T}^{\mu}_{\mu}\left[\!\left[\mathsf{int}\right]\!\right]_{\rho} &= \mathsf{int} \\
\mathbb{T}^{\mu}_{\mu}\left[\!\left[\tau_1 \xrightarrow{\varphi} \tau_2\right]\!\right]_{\rho} &= \Pi\vartheta \succeq \varphi.^{\rho}(\mathbb{T}^{\tau}_{\tau}\left[\!\left[\tau_1\right]\!\right] \xrightarrow{\vartheta} \mathbb{T}^{\tau}_{\tau}\left[\!\left[\tau_2\right]\!\right], \rho) \\
\mathbb{T}^{\mu}_{\mu}\left[\!\left[\tau_1 \times \tau_2\right]\!\right]_{\rho} &= \mathbb{T}^{\tau}_{\tau}\left[\!\left[\tau_1\right]\!\right] \times \mathbb{T}^{\tau}_{\tau}\left[\!\left[\tau_2\right]\!\right] \\
\mathbb{T}^{\mu}_{\mu}\left[\!\left[\Pi\varrho \succeq \varphi'.^{\varphi}\tau\right]\!\right]_{\rho} &= \Pi\varrho \succeq \varphi'.^{\rho}(\Pi\vartheta \succeq \varphi.^{\vartheta}\mathbb{T}^{\tau}_{\tau}\left[\!\left[\tau\right]\!\right], \rho)
\end{aligned}$$

Types

$$\begin{aligned}
\mathbb{T}^{\tau}_{\tau}\left[\!\left[\mathsf{bool}\right]\!\right] &= \mathsf{bool} \\
\mathbb{T}^{\tau}_{\tau}\left[\!\left[(\mu, \rho)\right]\!\right] &= \mathbb{T}^{\mu}_{\mu}\left[\!\left[\mu\right]\!\right]_{\rho}
\end{aligned}$$

Value contexts

$$\begin{aligned}
\mathbb{T}^{\Gamma}_{\Gamma}\left[\!\left[\cdot\right]\!\right] &= \cdot \\
\mathbb{T}^{\Gamma}_{\Gamma}\left[\!\left[\Gamma, x : \tau\right]\!\right] &= \mathbb{T}^{\Gamma}_{\Gamma}\left[\!\left[\Gamma\right]\!\right], x : \mathbb{T}^{\tau}_{\tau}\left[\!\left[\tau\right]\!\right]
\end{aligned}$$

Expressions

$$\begin{aligned}
\mathbb{T}^{e}_{e}\left[\!\left[i \text{ at } \rho\right]\!\right]_{\theta} &= i \text{ at } \rho \\
\mathbb{T}^{e}_{e}\left[\!\left[e_1 \oplus e_2 \text{ at } \rho\right]\!\right]_{\theta} &= \mathbb{T}^{e}_{e}\left[\!\left[e_1\right]\!\right]_{\theta} \oplus \mathbb{T}^{e}_{e}\left[\!\left[e_2\right]\!\right]_{\theta} \text{ at } \rho \\
\mathbb{T}^{e}_{e}\left[\!\left[e_1 \otimes e_2\right]\!\right]_{\theta} &= \mathbb{T}^{e}_{e}\left[\!\left[e_1\right]\!\right]_{\theta} \otimes \mathbb{T}^{e}_{e}\left[\!\left[e_2\right]\!\right]_{\theta} \\
\mathbb{T}^{e}_{e}\left[\!\left[\mathsf{tt}\right]\!\right]_{\theta} &= \mathsf{tt} \\
\mathbb{T}^{e}_{e}\left[\!\left[\mathsf{ff}\right]\!\right]_{\theta} &= \mathsf{ff} \\
\mathbb{T}^{e}_{e}\left[\!\left[\mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f\right]\!\right]_{\theta} &= \mathsf{if}\ \mathbb{T}^{e}_{e}\left[\!\left[e_b\right]\!\right]_{\theta}\ \mathsf{then}\ \mathbb{T}^{e}_{e}\left[\!\left[e_t\right]\!\right]_{\theta}\ \mathsf{else}\ \mathbb{T}^{e}_{e}\left[\!\left[e_f\right]\!\right]_{\theta} \\
\mathbb{T}^{e}_{e}\left[\!\left[x\right]\!\right]_{\theta} &= x \\
\mathbb{T}^{e}_{e}\left[\!\left[\lambda x : \tau.^{\varphi} e \text{ at } \rho\right]\!\right]_{\theta} &= \lambda\vartheta \succeq \varphi.^{\rho}(\lambda x : \mathbb{T}^{\tau}_{\tau}\left[\!\left[\tau\right]\!\right].^{\vartheta}\mathbb{T}^{e}_{e}\left[\!\left[e\right]\!\right]_{\vartheta} \text{ at } \rho) \text{ at } \rho \\
\mathbb{T}^{e}_{e}\left[\!\left[e_1\ e_2\right]\!\right]_{\theta} &= \mathbb{T}^{e}_{e}\left[\!\left[e_1\right]\!\right]\ [\theta]\ \mathbb{T}^{e}_{e}\left[\!\left[e_2\right]\!\right] \\
\mathbb{T}^{e}_{e}\left[\!\left[(e_1, e_2) \text{ at } \rho\right]\!\right]_{\theta} &= (\mathbb{T}^{e}_{e}\left[\!\left[e_1\right]\!\right]_{\theta}, \mathbb{T}^{e}_{e}\left[\!\left[e_2\right]\!\right]_{\theta}) \text{ at } \rho \\
\mathbb{T}^{e}_{e}\left[\!\left[\mathsf{fst}\ e\right]\!\right]_{\theta} &= \mathsf{fst}\ \mathbb{T}^{e}_{e}\left[\!\left[e\right]\!\right]_{\theta} \\
\mathbb{T}^{e}_{e}\left[\!\left[\mathsf{snd}\ e\right]\!\right]_{\theta} &= \mathsf{snd}\ \mathbb{T}^{e}_{e}\left[\!\left[e\right]\!\right]_{\theta} \\
\mathbb{T}^{e}_{e}\left[\!\left[\mathsf{letregion}\ \varrho.e\right]\!\right]_{\theta} &= \mathsf{letregion}\ \varrho.\mathbb{T}^{e}_{e}\left[\!\left[e\right]\!\right]_{\varrho} \\
\mathbb{T}^{e}_{e}\left[\!\left[\lambda\varrho \succeq \varphi'.^{\varphi} u \text{ at } \rho\right]\!\right]_{\theta} &= \lambda\varrho \succeq \varphi'.^{\rho}(\lambda\vartheta \succeq \varphi.^{\vartheta}\mathbb{T}^{e}_{e}\left[\!\left[u\right]\!\right]_{\vartheta} \text{ at } \rho) \text{ at } \rho \\
\mathbb{T}^{e}_{e}\left[\!\left[e\ [\rho]\right]\!\right]_{\theta} &= \mathbb{T}^{e}_{e}\left[\!\left[e\right]\!\right]_{\theta}\ [\rho]\ [\theta]
\end{aligned}$$

Programs

$$\mathbb{T}^{p}_{p}\left[\!\left[p\right]\!\right] = \mathbb{T}^{e}_{e}\left[\!\left[p\right]\!\right]_{\mathcal{H}}$$

Figure 19: Translation from the Bounded Effect Calculus to the Single Effect Calculus

Applying (3) and Lemma 12 to $\Delta \vdash_{\text{eff}} \varphi$ and $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{eff}} \varphi$.
Note that $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi$ for a suitably chosen ϑ. Furthermore, $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi \vdash_{\text{place}} \vartheta'$.
Applying Lemma 12 to $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{place}} \rho$ and $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi \vdash_{\text{place}} \rho$.
Applying the induction hypothesis to $\Delta \vdash_{\text{type}} \mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!]$ and $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi \vdash_{\text{type}} \mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!]$.
Applying the induction hypothesis to $\Delta \vdash_{\text{type}} \mathbb{T}^{\tau}_{\tau}[\![\tau_2]\!]$ and $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi \vdash_{\text{type}} \mathbb{T}^{\tau}_{\tau}[\![\tau_2]\!]$.

Case $\dfrac{\Delta \vdash_{\text{type}} \tau_1 \qquad \Delta \vdash_{\text{type}} \tau_2}{\Delta \vdash_{\text{btype}} \tau_1 \times \tau_2}$: We are required to show

$$\dfrac{\boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{type}} \mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!]} \qquad \boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{type}} \mathbb{T}^{\tau}_{\tau}[\![\tau_2]\!]}}{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{btype}} \mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!] \times \mathbb{T}^{\tau}_{\tau}[\![\tau_2]\!]}$$

Applying the induction hypothesis to $\Delta \vdash_{\text{type}} \mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!]$ and $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{type}} \mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!]$.
Applying the induction hypothesis to $\Delta \vdash_{\text{type}} \mathbb{T}^{\tau}_{\tau}[\![\tau_2]\!]$ and $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{type}} \mathbb{T}^{\tau}_{\tau}[\![\tau_2]\!]$.

Case $\dfrac{\Delta \vdash_{\text{eff}} \varphi' \qquad \Delta, \varrho \succeq \varphi' \vdash_{\text{eff}} \varphi \qquad \Delta, \varrho \succeq \varphi' \vdash_{\text{type}} \tau}{\Delta \vdash_{\text{btype}} \Pi \varrho \succeq \varphi'.^{\varphi} \tau}$: We are required to show

$$\dfrac{\boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{eff}} \varphi'} \qquad \boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi' \vdash_{\text{place}} \rho} \qquad \dfrac{\dfrac{\boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi' \vdash_{\text{eff}} \varphi} \qquad \dfrac{\boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi', \vartheta \succeq \varphi \vdash_{\text{place}} \vartheta} \qquad \boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi', \vartheta \succeq \varphi \vdash_{\text{type}} \mathbb{T}^{\tau}_{\tau}[\![\tau]\!]}}{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi' \vdash_{\text{btype}} \Pi \vartheta \succeq \varphi.^{\vartheta} \mathbb{T}^{\tau}_{\tau}[\![\tau]\!]} \qquad \boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi' \vdash_{\text{place}} \rho}}{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi' \vdash_{\text{type}} (\Pi \vartheta \succeq \varphi.^{\vartheta} \mathbb{T}^{\tau}_{\tau}[\![\tau]\!], \rho)}}{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{btype}} \Pi \varrho \succeq \varphi'.^{\rho} (\Pi \vartheta \succeq \varphi.^{\vartheta} \mathbb{T}^{\tau}_{\tau}[\![\tau]\!], \rho)}$$

Applying (3) and Lemma 12 to $\Delta \vdash_{\text{eff}} \varphi'$ and $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{eff}} \varphi'$.
Note that $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi'$ (equivalently, $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \varrho \succeq \varphi', \Delta'$) for a suitably chosen ϱ.
Applying Lemma 12 to $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{place}} \rho$ and $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi'$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi' \vdash_{\text{place}} \rho$.
Applying (3) and Lemma 12 to $\Delta \vdash_{\text{eff}} \varphi$ and $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \varrho \succeq \varphi', \Delta'$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \varrho \succeq \varphi', \Delta' \vdash_{\text{eff}} \varphi$ (equivalently, $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi' \vdash_{\text{eff}} \varphi$).
Note that $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi', \vartheta \succeq \varphi$ (equivalently, $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \varrho \succeq \varphi', \Delta', \vartheta \succeq \varphi$) for a suitably chosen ϑ. Hence, $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi, \varrho \succeq \varphi' \vdash_{\text{place}} \vartheta$.
Applying the induction hypothesis to $\Delta, \varrho \vdash_{\text{type}} \mathbb{T}^{\tau}_{\tau}[\![\tau]\!]$ and $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \varrho \succeq \varphi', \Delta', \vartheta \succeq \varphi$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \varrho \succeq \varphi', \Delta', \vartheta \succeq \varphi \vdash_{\text{type}} \mathbb{T}^{\tau}_{\tau}[\![\tau]\!]$ (equivalently, $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi', \vartheta \succeq \varphi \vdash_{\text{type}} \mathbb{T}^{\tau}_{\tau}[\![\tau]\!]$).

Case $\dfrac{\vdash_{\text{rctxt}} \Delta}{\Delta \vdash_{\text{type}} \mathsf{bool}}$: We are required to show

$$\dfrac{\boxed{\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \ \checkmark}}{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{type}} \mathsf{bool}}$$

Case $\dfrac{\Delta \vdash_{\text{btype}} \mu \qquad \Delta \vdash_{\text{place}} \rho}{\Delta \vdash_{\text{type}} (\mu, \rho)}$: We are required to show

$$\dfrac{\boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{btype}} \mathbb{T}^{\mu}_{\mu}[\![\mu]\!]_{\rho}} \qquad \boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{place}} \rho}}{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{type}} (\mathbb{T}^{\mu}_{\mu}[\![\mu]\!]_{\rho}, \rho)}$$

Applying (2) and Lemma 12 to $\Delta \vdash_{\text{place}} \rho$ and $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{place}} \rho$.
Applying the induction hypothesis to $\Delta \vdash_{\text{btype}} \mu$ and $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{place}} \rho$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{btype}} \mathbb{T}^{\mu}_{\mu}[\![\mu]\!]_{\rho}$.

(6) By induction on the derivation $\Delta \vdash_{\text{vctxt}} \Gamma$.

(7) Proceed by induction on the derivation $\Delta; \Gamma \vdash_{\text{exp}} e : \tau, \varphi$. Note that by applying Lemma 6 to $\vdash_{\text{ctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!]; \theta$, we conclude that $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!]; \theta$, $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{vctxt}} \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!]$, and $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{place}} \theta$.
Note that if $\Delta \vdash_{\text{place}} \rho$ then by (2) and Lemma 12, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{place}} \rho$.
Note that if $\Delta \vdash_{\text{eff}} \varphi$ then by (3) and Lemma 12, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{eff}} \varphi$.
Note that if $\Delta \vdash_{\text{rr}} \rho \succeq \rho'$ then by Lemma 12, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{rr}} \rho \succeq \rho''$.
Note that if $\Delta \vdash_{\text{re}} \rho \succeq \varphi$ then by Lemma 12, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{re}} \rho \succeq \varphi$.
Note that if $\Delta \vdash_{\text{er}} \varphi \ni \rho$, then by $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{re}} \theta \succeq \varphi$, $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{rr}} \theta \succeq \rho$.
Note that if $\Delta \vdash_{\text{ee}} \varphi \supseteq \varphi'$, then by $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{re}} \theta \succeq \varphi$, $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{re}} \theta \succeq \varphi'$.

Case $\dfrac{\begin{array}{c}\vdash_{\text{ctxt}} \Delta; \Gamma; \varphi \\ \Delta \vdash_{\text{place}} \rho \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho\end{array}}{\Delta; \Gamma \vdash_{\text{exp}} i \text{ at } \rho : (\mathsf{int}, \rho), \varphi}$: We are required to show:

$$\dfrac{\boxed{\vdash_{\text{ctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!]; \theta}^{\checkmark} \qquad \boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{place}} \rho}^{\checkmark} \qquad \boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{rr}} \theta \succeq \rho}^{\checkmark}}{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} i \text{ at } \rho : (\mathsf{int}, \rho), \theta}$$

Case $\dfrac{\begin{array}{c}\Delta; \Gamma \vdash_{\text{exp}} e_1 : (\mathsf{int}, \rho_1), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho_1 \\ \Delta; \Gamma \vdash_{\text{exp}} e_2 : (\mathsf{int}, \rho_2), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho_2 \\ \Delta \vdash_{\text{place}} \rho \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho\end{array}}{\Delta; \Gamma \vdash_{\text{exp}} e_1 \oplus e_2 \text{ at } \rho : (\mathsf{int}, \rho), \varphi}$: We are required to show

$$\dfrac{\begin{array}{cc}\boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \mathbb{T}^{e}_{e}[\![e_1]\!]_{\theta} : (\mathsf{int}, \rho_1), \theta} & \boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{rr}} \theta \succeq \rho_1}^{\checkmark} \\ \boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \mathbb{T}^{e}_{e}[\![e_2]\!]_{\theta} : (\mathsf{int}, \rho_2), \theta} & \boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{rr}} \theta \succeq \rho_2}^{\checkmark} \\ \boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{place}} \rho}^{\checkmark} & \boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{rr}} \theta \succeq \rho}^{\checkmark}\end{array}}{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \mathbb{T}^{e}_{e}[\![e_1]\!]_{\theta} \oplus \mathbb{T}^{e}_{e}[\![e_2]\!]_{\theta} \text{ at } \rho : (\mathsf{int}, \rho), \theta}$$

Applying the induction hypothesis to $\Delta; \Gamma \vdash_{\text{exp}} e_1 : (\mathsf{int}, \rho_1), \varphi$, $\vdash_{\text{ctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!]; \theta$, and $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{re}} \theta \succeq \varphi$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \mathbb{T}^{e}_{e}[\![e_1]\!]_{\theta} : (\mathsf{int}, \rho_1), \theta$.
Applying the induction hypothesis to $\Delta; \Gamma \vdash_{\text{exp}} e_2 : (\mathsf{int}, \rho_2), \varphi$, $\vdash_{\text{ctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!]; \theta$, and $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{re}} \theta \succeq \varphi$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \mathbb{T}^{e}_{e}[\![e_2]\!]_{\theta} : (\mathsf{int}, \rho_2), \theta$.

Case $\dfrac{\begin{array}{c}\Delta; \Gamma \vdash_{\text{exp}} e_1 : (\mathsf{int}, \rho_1), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho_1 \\ \Delta; \Gamma \vdash_{\text{exp}} e_2 : (\mathsf{int}, \rho_2), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho_2\end{array}}{\Delta; \Gamma \vdash_{\text{exp}} e_1 \otimes e_2 : \mathsf{bool}, \varphi}$: ...

Case $\dfrac{\vdash_{\text{ctxt}} \Delta; \Gamma; \varphi}{\Delta; \Gamma \vdash_{\text{exp}} \mathsf{tt} : \mathsf{bool}, \varphi}$: ...

Case $\dfrac{\vdash_{\text{ctxt}} \Delta; \Gamma; \varphi}{\Delta; \Gamma \vdash_{\text{exp}} \mathsf{ff} : \mathsf{bool}, \varphi}$: ...

Case $\dfrac{\begin{array}{c}\Delta; \Gamma \vdash_{\text{exp}} e_b : \mathsf{bool}, \varphi \\ \Delta; \Gamma \vdash_{\text{exp}} e_t : \tau, \varphi \\ \Delta; \Gamma \vdash_{\text{exp}} e_f : \tau, \varphi\end{array}}{\Delta; \Gamma \vdash_{\text{exp}} \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f : \tau, \varphi}$: ...

Case $\dfrac{\vdash_{\text{ctxt}} \Delta;\Gamma;\varphi \qquad x \in dom(\Gamma) \qquad \Gamma(x) = \tau}{\Delta;\Gamma \vdash_{\text{exp}} x : \tau, \varphi}$: ...

Case $\dfrac{\Delta;\Gamma, x:\tau_1 \vdash_{\text{exp}} e' : \tau_2, \varphi' \qquad \Delta \vdash_{\text{place}} \rho \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho}{\Delta;\Gamma \vdash_{\text{exp}} \lambda x:\tau_1.^{\varphi'} e' \text{ at } \rho : (\tau_1 \xrightarrow{\varphi'} \tau_2, \rho), \varphi}$: We are required to show

$$\dfrac{\dfrac{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!], x : \mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!] \vdash_{\text{exp}} \mathbb{T}^{e}_{e}[\![e']\!]_{\vartheta} : \mathbb{T}^{\tau}_{\tau}[\![\tau_2]\!], \vartheta \qquad \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi' \vdash_{\text{place}} \rho \qquad \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi' \vdash_{\text{rr}} \rho \succeq \rho}{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \lambda x : \mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!].^{\vartheta} \mathbb{T}^{e}_{e}[\![e']\!]_{\vartheta} \text{ at } \rho : (\mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!] \xrightarrow{\vartheta} \mathbb{T}^{\tau}_{\tau}[\![\tau_2]\!], \rho), \rho} \qquad \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{place}} \rho \checkmark \qquad \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{rr}} \theta \succeq \rho \checkmark}{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \lambda\vartheta \succeq \varphi'.^{\rho}(\lambda x : \mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!].^{\vartheta} \mathbb{T}^{e}_{e}[\![e']\!]_{\vartheta} \text{ at } \rho) \text{ at } \rho : (\Pi\vartheta \succeq \varphi'.^{\rho}(\mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!] \xrightarrow{\vartheta} \mathbb{T}^{\tau}_{\tau}[\![\tau_2]\!], \rho), \rho), \theta}$$

Note that $\Delta;\Gamma, x:\tau_1 \vdash_{\text{exp}} e' : \tau_2, \varphi'$ implies $\Delta \vdash_{\text{eff}} \varphi'$ (by an unproven well-formedness lemma). Applying (3) and Lemma 12 to $\Delta \vdash_{\text{eff}} \varphi'$ and $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{eff}} \varphi'$. Note that $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi'$ for a suitably chosen ϑ.
Applying (2) and Lemma 12 to $\Delta \vdash_{\text{place}} \rho$ and $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi'$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi' \vdash_{\text{place}} \rho$ and $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi' \vdash_{\text{re}} \rho \succeq \rho$.
Note that $\Delta;\Gamma, x:\tau_1 \vdash_{\text{exp}} e' : \tau_2, \varphi'$ implies $\Delta \vdash_{\text{vctxt}} \Gamma, x:\tau_1$ (by an unproven well-formedness lemma). Applying (6) and Lemma 12 to $\Delta \vdash_{\text{vctxt}} \Gamma, x:\tau_1$ and $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi'$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi' \vdash_{\text{vctxt}} \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!], x : \mathbb{T}^{\tau}_{\tau}[\![\tau]\!]$.
Applying the induction hypothesis to $\Delta;\Gamma, x:\tau_1 \vdash_{\text{exp}} e' : \tau_2, \varphi'$, $\vdash_{\text{ctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!], x : \mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!]; \vartheta$, and $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi' \vdash_{\text{re}} \vartheta \succeq \varphi'$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!], x : \mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!] \vdash_{\text{exp}} \mathbb{T}^{e}_{e}[\![e']\!]_{\vartheta} : \mathbb{T}^{\tau}_{\tau}[\![\tau_2]\!], \vartheta$.

Case $\dfrac{\Delta;\Gamma \vdash_{\text{exp}} e_1 : (\tau_1 \xrightarrow{\varphi'} \tau_2, \rho'_1), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho'_1 \qquad \Delta;\Gamma \vdash_{\text{exp}} e_2 : \tau_1, \varphi \qquad \Delta \vdash_{\text{ee}} \varphi \supseteq \varphi'}{\Delta;\Gamma \vdash_{\text{exp}} e_1\ e_2 : \tau_2, \varphi}$: Note that $\Delta;\Gamma \vdash_{\text{exp}} e_1 : (\tau_1 \xrightarrow{\varphi'} \tau_2, \rho'_1), \varphi$ implies $\Delta \vdash_{\text{type}} \tau_1$ and $\Delta \vdash_{\text{type}} \tau_2$ (by an unproven well-formedness lemma).
Hence, for $\vartheta \notin dom(\Delta)$, $\tau_1[\theta/\vartheta] \equiv \tau_1$ and $\tau_2[\theta/\vartheta] \equiv \tau_2$. Likewise, $\mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!][\theta/\vartheta] \equiv \mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!]$ and $\mathbb{T}^{\tau}_{\tau}[\![\tau_2]\!][\theta/\vartheta] \equiv \mathbb{T}^{\tau}_{\tau}[\![\tau_2]\!]$.
We are required to show

$$\dfrac{\dfrac{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \mathbb{T}^{e}_{e}[\![e_1]\!]_{\theta} : (\Pi\vartheta \succeq \varphi'.^{\rho'_1}(\mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!] \xrightarrow{\vartheta} \mathbb{T}^{\tau}_{\tau}[\![\tau_2]\!], \rho'_1), \rho'_1), \theta \qquad \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{rr}} \theta \succeq \rho'_1 \checkmark \qquad \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{place}} \theta \checkmark \qquad \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{re}} \theta \succeq \varphi' \checkmark \qquad \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{rr}} \theta \succeq \rho'_1 \checkmark}{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \mathbb{T}^{e}_{e}[\![e_1]\!]_{\theta}\ [\theta] : (\mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!] \xrightarrow{\theta} \mathbb{T}^{\tau}_{\tau}[\![\tau_2]\!], \rho'_1), \theta} \qquad \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{rr}} \theta \succeq \rho'_1 \checkmark \qquad \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \mathbb{T}^{e}_{e}[\![e_1]\!]_{\theta} : \mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!], \theta \qquad \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{rr}} \theta \succeq \theta \checkmark}{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \mathbb{T}^{e}_{e}[\![e_1]\!]_{\theta}\ [\theta]\ \mathbb{T}^{e}_{e}[\![e_2]\!]_{\theta} : \mathbb{T}^{\tau}_{\tau}[\![\tau_2]\!], \theta}$$

Applying the induction hypothesis to $\Delta;\Gamma \vdash_{\text{exp}} e_1 : (\tau_1 \xrightarrow{\varphi'} \tau_2, \rho'_1), \varphi$, $\vdash_{\text{ctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!]; \theta$, and $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{re}} \theta \succeq \varphi$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \mathbb{T}^{e}_{e}[\![e_1]\!]_{\theta} : (\Pi\vartheta \succeq \varphi'.^{\rho'_1}(\mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!] \xrightarrow{\vartheta} \mathbb{T}^{\tau}_{\tau}[\![\tau_2]\!], \rho'_1), \rho'_1), \theta$.
Applying the induction hypothesis to $\Delta;\Gamma \vdash_{\text{exp}} e_2 : \tau_1, \varphi$, $\vdash_{\text{ctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!]; \theta$, and $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{re}} \theta \succeq \varphi$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \mathbb{T}^{e}_{e}[\![e_2]\!]_{\theta} : \mathbb{T}^{\tau}_{\tau}[\![\tau_1]\!], \theta$.

Case $\dfrac{\Delta;\Gamma \vdash_{\text{exp}} e_1 : \tau_1, \varphi \qquad \Delta;\Gamma \vdash_{\text{exp}} e_2 : \tau_2, \varphi \qquad \Delta \vdash_{\text{place}} \rho \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho}{\Delta;\Gamma \vdash_{\text{exp}} (e_1, e_2) \text{ at } \rho : (\tau_1 \times \tau_2, \rho), \varphi}$: ...

Case $\dfrac{\Delta;\Gamma \vdash_{\text{exp}} e : (\tau_1 \times \tau_2, \rho), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho}{\Delta;\Gamma \vdash_{\text{exp}} \mathsf{fst}\ \ e : \tau_1, \varphi}$: ...

Case $\dfrac{\begin{array}{c}\Delta;\Gamma \vdash_{\text{exp}} e : (\tau_1 \times \tau_2, \rho), \varphi \\ \Delta \vdash_{\text{er}} \varphi \ni \rho\end{array}}{\Delta;\Gamma \vdash_{\text{exp}} \mathsf{snd}\ \ e : \tau_2, \varphi}$: ...

Case $\dfrac{\begin{array}{c}\Delta \vdash_{\text{type}} \tau \qquad \vdash_{\text{ctxt}} \Delta;\Gamma;\{\rho_1,\ldots,\rho_n\} \\ \Delta, \varrho \succeq \{\rho_1,\ldots,\rho_n\};\Gamma \vdash_{\text{exp}} e : \tau, \{\rho_1,\ldots,\rho_n,\varrho\}\end{array}}{\Delta;\Gamma \vdash_{\text{exp}} \mathsf{letregion}\ \varrho.e : \tau, \{\rho_1,\ldots,\rho_n\}}$: We are required to show

$$\dfrac{\vdash_{\text{ctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta';\mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!];\theta\ ^{\checkmark} \qquad \dfrac{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{type}} \mathbb{T}^{\tau}_{\tau}[\![\tau]\!] \qquad \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \{\theta\};\mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \mathbb{T}^{e}_{e}[\![e]\!]_{\varrho} : \mathbb{T}^{\tau}_{\tau}[\![\tau]\!], \varrho}{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta';\mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \mathsf{letregion}\ \varrho.\mathbb{T}^{e}_{e}[\![e]\!]_{\varrho} : \mathbb{T}^{\tau}_{\tau}[\![\tau]\!], \theta}$$

Applying (5) to $\Delta \vdash_{\text{type}} \tau$ and $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{type}} \mathbb{T}^{\tau}_{\tau}[\![\tau]\!]$.
Note that $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \{\theta\}$ for a suitably chosen ϱ.
Applying Lemma 12 to $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{vctxt}} \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!]$ and $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \{\theta\}$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \{\theta\} \vdash_{\text{vctxt}} \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!]$.
Clearly, $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \{\theta\} \vdash_{\text{place}} \varrho$. Hence, $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \{\theta\};\mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!];\varrho$.
Notet that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \{\theta\} \vdash_{\text{re}} \varrho \succeq \{\rho_1,\ldots,\rho_n,\varrho\}$, by transitivity with θ in the case of ρ_i and by reflexivity in the case of ϱ.
Applying the induction hypothesis to $\Delta, \varrho \succeq \{\rho_1,\ldots,\rho_n\};\Gamma \vdash_{\text{exp}} e : \tau, \{\rho_1,\ldots,\rho_n,\varrho\}$, $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \{\theta\};\mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!];\varrho$, and $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \{\theta\} \vdash_{\text{re}} \varrho \succeq \{\rho_1,\ldots,\rho_n,\varrho\}$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \{\theta\};\mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \mathbb{T}^{e}_{e}[\![e]\!]_{\varrho} : \mathbb{T}^{\tau}_{\tau}[\![\tau]\!], \varrho$.

Case $\dfrac{\begin{array}{c}\Delta, \varrho \succeq \varphi'';\Gamma \vdash_{\text{exp}} u' : \tau, \varphi' \\ \Delta \vdash_{\text{place}} \rho \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho\end{array}}{\Delta;\Gamma \vdash_{\text{exp}} \lambda\varrho \succeq \varphi''.^{\varphi'} u' \ \mathsf{at}\ \rho : (\Pi\varrho.^{\varphi'}\tau, \rho), \varphi}$: We are required to show

$$\dfrac{\dfrac{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi'', \vartheta \succeq \varphi';\mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \mathbb{T}^{e}_{e}[\![u']\!]_{\vartheta} : \mathbb{T}^{\tau}_{\tau}[\![\tau]\!], \vartheta \qquad \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi'' \vdash_{\text{rr}} \rho \succeq \rho}{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \{\};\mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \lambda\vartheta \succeq \varphi'.^{\vartheta}\mathbb{T}^{e}_{e}[\![u']\!]_{\vartheta}\ \mathsf{at}\ \rho : (\Pi\vartheta \succeq \varphi'.^{\vartheta}\mathbb{T}^{\tau}_{\tau}[\![\tau]\!], \rho), \rho} \qquad \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{place}} \rho\ ^{\checkmark} \qquad \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{re}} \theta \succeq \rho\ ^{\checkmark}}{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta';\mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \lambda\varrho \succeq \varphi''.^{\rho}(\lambda\vartheta \succeq \varphi'.^{\vartheta}\mathbb{T}^{e}_{e}[\![u']\!]_{\vartheta}\ \mathsf{at}\ \rho)\ \mathsf{at}\ \rho : \Pi\varrho \succeq \varphi''.^{\rho}(\Pi\vartheta \succeq \varphi'.^{\vartheta}\mathbb{T}^{\tau}_{\tau}[\![\tau]\!], \rho), \theta}$$

Note that $\Delta, \varrho \succeq \varphi'';\Gamma \vdash_{\text{exp}} u' : \tau, \varphi'$ implies $\Delta \vdash_{\text{eff}} \varphi$ (by an unproven well-formedness lemma). Applying (3) and Lemma 12 to $\Delta \vdash_{\text{eff}} \varphi'$ and $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\text{eff}} \varphi'$.
Note that $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi''$ (equivalently, $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \varrho \succeq \varphi'', \Delta'$) for a suitably chosen ϱ.
Applying (2) and Lemma 12 to $\Delta \vdash_{\text{place}} \rho$ and $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi''$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi'' \vdash_{\text{place}} \rho$ and $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi'' \vdash_{\text{re}} \rho \succeq \rho$.
Note that $\Delta, \varrho \succeq \varphi'';\Gamma \vdash_{\text{exp}} u' : \tau, \varphi'$ implies $\Delta \vdash_{\text{eff}} \varphi'$ (by an unproven well-formedness lemma). Applying (3) and Lemma 12 to $\Delta \vdash_{\text{eff}} \varphi'$ and $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi''$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi'' \vdash_{\text{eff}} \varphi'$. Note that $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \varrho \succeq \varphi'', \vartheta \succeq \varphi'$ (equivalently, $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \varrho \succeq \varphi'', \Delta', \vartheta \succeq \varphi'$) for a suitably chosen ϑ.
Applying Lemma 12 to $\Delta \vdash_{\text{vctxt}} \Gamma$ and $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \varrho \succeq \varphi'', \Delta', \vartheta \succeq \varphi'$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \varrho \succeq \varphi'', \Delta', \vartheta \succeq \varphi' \vdash_{\text{vctxt}} \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!]$.
Applying the induction hypothesis to $\Delta, \varrho \succeq \varphi'';\Gamma \vdash_{\text{exp}} u' : \tau, \varphi'$, $\vdash_{\text{ctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \varrho \succeq \varphi'', \Delta', \vartheta \succeq \varphi';\mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!];\vartheta$, and $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \varrho \succeq \varphi'', \Delta', \vartheta \succeq \varphi' \vdash_{\text{re}} \vartheta \succeq \varphi'$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \varrho \succeq \varphi'', \Delta', \vartheta \succeq \varphi';\mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \mathbb{T}^{e}_{e}[\![u']\!]_{\vartheta} : \mathbb{T}^{\tau}_{\tau}[\![\tau]\!], \vartheta$ (equivalently, $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta', \vartheta \succeq \varphi', \varrho \succeq \varphi'';\mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\text{exp}} \mathbb{T}^{e}_{e}[\![u']\!]_{\vartheta} : \mathbb{T}^{\tau}_{\tau}[\![\tau]\!], \vartheta$).

Case $\dfrac{\begin{array}{c}\Delta;\Gamma \vdash_{\text{exp}} e : (\Pi\varrho \succeq \varphi''.^{\varphi'}\tau, \rho'_1), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho'_1 \\ \Delta \vdash_{\text{place}} \rho_2 \qquad \Delta \vdash_{\text{re}} \rho_2 \succeq \varphi'' \qquad \Delta \vdash_{\text{ee}} \varphi \supseteq \varphi'[\rho_2/\varrho]\end{array}}{\Delta;\Gamma \vdash_{\text{exp}} e\ [\rho_2] : \tau[\rho_2/\varrho], \varphi}$: Note that $\Delta;\Gamma \vdash_{\text{exp}} e_1 : (\Pi\varrho \succeq \varphi''.\tau, \rho'_1), \varphi$ implies $\Delta \vdash_{\text{type}} (\Pi\varrho \succeq \varphi''.\tau, \rho'_1)$ (by an unproven well-formedness lemma).

Hence, for $\Delta \vdash_{\mathrm{re}} \rho_2 \succeq \varphi''$ and $\vartheta \notin dom(\Delta)$, $\tau[\rho_2/\varrho][\theta/\vartheta] \equiv \tau[\rho_2/\varrho]$. Likewise $\mathbb{T}^{\tau}_{\tau}[\![\tau]\!][\rho_2/\varrho][\theta/\vartheta] \equiv \mathbb{T}^{\tau}_{\tau}[\![\tau[\rho_2/\varrho]]\!][\theta/\vartheta] \equiv \mathbb{T}^{\tau}_{\tau}[\![\tau[\rho_2/\varrho][\theta/\vartheta]]\!] \equiv \mathbb{T}^{\tau}_{\tau}[\![\tau[\rho_2/\varrho]]\!]$.
Note that $(\Pi\vartheta \succeq \varphi.^{\vartheta}\mathbb{T}^{\tau}_{\tau}[\![\tau]\!])[\rho_2/\varrho] \equiv \Pi\vartheta \succeq \varphi[\rho_2/\varrho].^{\vartheta}\mathbb{T}^{\tau}_{\tau}[\![\tau]\!][\rho_2/\varrho] \equiv \Pi\vartheta \succeq \varphi[\rho_2/\varrho].^{\vartheta}\mathbb{T}^{\tau}_{\tau}[\![\tau[\rho_2/\varrho]]\!]$.
We are required to show

$$\dfrac{\dfrac{\boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\mathrm{exp}} \mathbb{T}^{e}_{e}[\![e]\!]_{\theta} : (\Pi\varrho \succeq \varphi''.^{\rho'_1}(\Pi\vartheta \succeq \varphi'.^{\vartheta}\mathbb{T}^{\tau}_{\tau}[\![\tau]\!], \rho'_1), \rho'_1), \theta} \quad \boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\mathrm{rr}} \theta \succeq \rho'_1\ {}^{\checkmark}}}{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\mathrm{exp}} \mathbb{T}^{e}_{e}[\![e]\!]_{\theta}\ \ [\rho_2] : (\Pi\vartheta \succeq \varphi'[\rho_2/\varrho].^{\vartheta}\mathbb{T}^{\tau}_{\tau}[\![\tau[\rho_2/\varrho]]\!], \rho'_1), \theta} \quad \boxed{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\mathrm{rr}} \theta \succeq \rho'_1\ {}^{\checkmark}}}{\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\mathrm{exp}} \mathbb{T}^{e}_{e}[\![e]\!]_{\theta}\ \ [\rho_2]\ [\theta] : \mathbb{T}^{\tau}_{\tau}[\![\tau[\rho_2/\varrho]]\!], \theta}$$

where the boxed premises are $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\mathrm{place}} \rho_2\ {}^{\checkmark}$, $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\mathrm{re}} \rho_2 \succeq \varphi''\ {}^{\checkmark}$, $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\mathrm{rr}} \theta \succeq \rho'_1\ {}^{\checkmark}$, $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\mathrm{place}} \theta\ {}^{\checkmark}$, $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\mathrm{re}} \theta \succeq \varphi'[\rho_2/\varrho]\ {}^{\checkmark}$, and $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta' \vdash_{\mathrm{rr}} \theta \succeq \theta\ {}^{\checkmark}$.

Applying the induction hypothesis to $\Delta; \Gamma \vdash_{\mathrm{exp}} e : (\Pi\varrho.^{\varphi'}\tau, \rho'_1), \varphi, \vdash_{\mathrm{ctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!]; \theta$, and $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!] \vdash_{\mathrm{re}} \theta \succeq \varphi$, we conclude that $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma}[\![\Gamma]\!] \vdash_{\mathrm{exp}} \mathbb{T}^{e}_{e}[\![e]\!]_{\theta} : (\Pi\varrho \succeq \varphi''.^{\rho'_1}(\Pi\vartheta \succeq \varphi'.^{\vartheta}\mathbb{T}^{\tau}_{\tau}[\![\tau]\!], \rho'_1), \rho'_1), \theta$.

(8) Proceed by inspection on the derivation $\vdash_{\mathrm{prog}} p\ \mathsf{ok}$.

Case $\dfrac{\cdot, \mathcal{H} \succeq \{\}; \cdot \vdash_{\mathrm{exp}} p : \mathsf{bool}, \{\mathcal{H}\}}{\vdash_{\mathrm{prog}} p\ \mathsf{ok}}$:

We are required to show

$$\dfrac{\cdot, \mathcal{H} \succeq \{\}; \cdot \vdash_{\mathrm{exp}} \mathbb{T}^{e}_{e}[\![p]\!]_{\mathcal{H}} : \mathsf{bool}, \mathcal{H}}{\vdash_{\mathrm{prog}} \mathbb{T}^{e}_{e}[\![p]\!]_{\mathcal{H}}\ \mathsf{ok}}$$

Applying (7) to $\cdot, \mathcal{H} \succeq \{\}; \cdot \vdash_{\mathrm{exp}} p : \mathsf{bool}, \{\mathcal{H}\}, \vdash_{\mathrm{ctxt}} \cdot, \mathcal{H} \succeq \{\}; \cdot; \mathcal{H}$, and $\cdot, \mathcal{H} \succeq \{\} \vdash_{\mathrm{re}} \mathcal{H} \succeq \{\mathcal{H}\}$, we conclude that $\cdot, \mathcal{H} \succeq \{\}; \cdot \vdash_{\mathrm{exp}} \mathbb{T}^{e}_{e}[\![p]\!]_{\mathcal{H}} : \mathbb{T}^{\tau}_{\tau}[\![\mathsf{bool}]\!], \mathcal{H}$.

□

$$
\begin{array}{rrcl}
 & i & \in & \mathbb{Z} \\
 & \alpha, \beta & \in & TVars^{FRGN} \\
 & f, x & \in & Vars^{FRGN} \\
 & & & \\
\text{Surface types} & \tau & ::= & \mathsf{int} \mid \mathsf{bool} \mid \tau_1 \to \tau_2 \mid \tau_1 \times \cdots \times \tau_n \mid \alpha \mid \forall \alpha.\tau \mid \\
 & & & \mathsf{RGN}\ \tau_r\ \tau_a \mid \mathsf{RGNVar}\ \tau_r\ \tau_a \mid \mathsf{RGNHandle}\ \tau_r \\
 & & & \\
\text{Surface terms} & e & ::= & i \mid e_1 \oplus e_2 \mid e_1 \otimes e_3 \mid \\
 & & & \mathsf{tt} \mid \mathsf{ff} \mid \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f \mid \\
 & & & x \mid \lambda x : \tau.e \mid e_1\ e_2 \mid \\
 & & & (e_1, \ldots, e_n) \mid \mathsf{sel}_i\ e \mid \\
 & & & \Lambda \alpha.e \mid e\ [\tau] \mid \\
 & & & \mathsf{let}\ x = e_1\ \mathsf{in}\ e_2 \mid \\
 & & & \mathsf{runRGN}\ [\tau_a]\ v \mid \kappa^v \\
\text{Surface commands} & \kappa^v & ::= & \underline{\mathsf{returnRGN}}\ [\tau_r]\ [\tau_a]\ v \mid \underline{\mathsf{thenRGN}}\ [\tau_r]\ [\tau_a]\ [\tau_b]\ v_1\ v_2 \mid \\
 & & & \underline{\mathsf{letRGN}}\ [\tau_r]\ [\tau_a]\ v \mid \\
 & & & \underline{\mathsf{newRGNVar}}\ [\tau_r]\ [\tau_a]\ v_1\ v_2 \mid \underline{\mathsf{readRGNVar}}\ [\tau_r]\ [\tau_a]\ v \mid \\
\text{Surface values} & v & ::= & i \mid \mathsf{tt} \mid \mathsf{ff} \mid x \mid \lambda x : \tau.e \mid (v_1, \ldots, v_n) \mid \Lambda \alpha.e \mid \kappa^v
\end{array}
$$

Figure 20: Surface syntax of $\mathsf{F}^{\mathsf{RGN}}$

3 The Target Calculus: $\mathsf{F}^{\mathsf{RGN}}$

The language $\mathsf{F}^{\mathsf{RGN}}$ is an extension of System F [20, 7] (also referred to as the polymorphic λ-calculus), adding monadic types and operations and taking inspiration from the work on monadic state [14, 15, 16, 1, 23, 19]. Essentially, $\mathsf{F}^{\mathsf{RGN}}$ uses an explicit region monad to enforce the locality of region allocated values.

We present the full formal language $\mathsf{F}^{\mathsf{RGN}}$ and a syntactic proof of type soundness. We begin with a presentation of the language (surface syntax, computation syntax, dynamic semantics, and static semantics) and then proceed to the proof.

The dynamic semantics defines a large-step (or natural) semantics, which defines an *evaluation relation* from *towers of stacks of regions* and *expressions* to *values*. Although the language presented here is strongly normalizing, we adopt a proof method using *natural transition semantics*, which models program execution in terms of transitions between *partial derivations*. This proof method can be extended in a straight-forward manner to handle non-terminating executions, as will arise from adding an fixRGNVar command.

The proof of type soundness is presented in "bottom-up" order, where all relevant lemmas are presented (and proved) before being used.

3.1 Surface Syntax of $\mathsf{F}^{\mathsf{RGN}}$

Figure 20 presents the syntax of "surface programs" (that is, excluding semantic values that will appear in the operational semantics) of $\mathsf{F}^{\mathsf{RGN}}$. In the following sections, we explain and motivate the main constructs of $\mathsf{F}^{\mathsf{RGN}}$.

3.1.1 Types

Types in $\mathsf{F}^{\mathsf{RGN}}$ are similar to those found in System F. We include the primitives types int and bool, function and product types, and type abstractions. In addition, we have $\mathsf{RGN}\ \tau_r\ \tau_a$ as the type of monadic region computations, $\mathsf{RGNVar}\ \tau_r\ \tau_a$ as the type of region allocated values, and $\mathsf{RGNHandle}\ \tau_r$ as the type of region handles. Intuitively, $\mathsf{RGN}\ \tau_r\ \tau_a$ is the type of computations that yield values of type τ_a and that take place in the region indexed by the type τ_r. Likewise, $\mathsf{RGNVar}\ \tau_r\ \tau_a$ is the type of values of type τ_a values allocated

in the region indexed by the type τ_r. Finally, RGNHandle τ_r is the type of handles for the region indexed by the type τ_r. A value of such a type is a *region handle* – a run-time value holding the data necessary to allocate values within a region. Region indices (types) and region handles (values) are distinguished in order to maintain a phase distinction between compile-time and run-time expressions and to more accurately reflect implementation of regions. Region indices, like other types, have no run-time significance and may be erased from compiled code. On the other hand, region handles are necessary at run-time to allocate values within a region.

Although surface programs will never require a region index to be represented by anything other than a type variable, we choose to allow an arbitrary type in the first argument of the RGN monad type constructor. We can thus interpret RGN as a primitive type constructor, without any special restrictions that may not be expressible in a practical programming language. Furthermore, in an operational semantics not based on type-erasure, such as that presented in the next section, type variables used as region indices will be instantiated with region types.

3.1.2 Terms

As with types, most of the terms in $\mathsf{F}^{\mathsf{RGN}}$ are similar to those found in System F. Constants, arithmetic and boolean operations, function abstraction and application, tuple introduction and elimination, and type abstraction and instantiation are all completely standard.

We let κ^v range over the syntactic class of monadic commands. (Equivalently, and as suggested by the explicit type annotations and the restriction of sub-expressions to values, we can consider the monadic commands as constants with polymorphic types in a call-by-value interpretation of $\mathsf{F}^{\mathsf{RGN}}$. Presenting monadic commands in this fashion avoids intermediate terms in the operational semantics corresponding to partial application.) Each command corresponds to a particular transformation on a monadic region. The commands returnRGN and thenRGN are the *unit* and *bind* operations of the region monad respectively.

The command newRGNVar $[\tau_r]$ $[\tau_a]$ v_r v_a allocates the value v_a of type τ_a in the region indexed by the type τ_r. The additional value v_r is the region handle for the region indexed by τ_r, which is necessary to allocate values within the region.

The command readRGNVar $[\tau_r]$ $[\tau_a]$ v_l reads a value of type τ_a stored at the location v_l in the region indexed by the type τ_r.

The command letRGN $[\tau_r]$ $[\tau_a]$ v first creates a new region, executes the region computation described by v in the new region, and finally deallocates the new region. This entire execution is a computation that yields a value of type τ_a taking place in the region indexed by τ_r. We will have more to say about the computation described by v shortly.

Finally, the expression runRGN $[\tau_a]$ v eliminates region-transformer operations. In particular, if v describes a region computation yielding a value of type τ_a, then runRGN $[\tau_a]$ v executes that computation in a region, returning the final value produced by v and destroying the region (and any values allocated within it). The region (and any new regions introduced by letRGN) is neither accessible nor visible from outside the runRGN $[\tau_a]$ v expression.

3.2 Computation Syntax of $\mathsf{F}^{\mathsf{RGN}}$

Figure 21 presents the synatx of "computation programs", which extends the syntax of the previous section with semantic values that appear in the operational semantics.

Stack names, region names, and locations are used to represent pointers to region allocated data. Because runRGN computations can be nested, we need a means to distinguish data allocated in regions that belong to different runRGN computations; stack names serve this purpose. Each runRGN computation is associated with a unique stack, which collects and identifies all regions belonging to that computation. Stack and region placeholders distinguish between live and dead stacks and regions; a dead stack or region corresponds to a deallocated stack or region.

The computation syntax adds one new type form. The type $\sigma\sharp\rho$ is the runtime representation of a region index. Such a type identifies the stack and region in which a monadic region computation is executing.

$$
\begin{array}{lrcl}
 & i & \in & \mathbb{Z} \\
 & \alpha, \varrho & \in & \mathit{TVars}^{FRGN} \\
 & f, x & \in & \mathit{Vars}^{FRGN} \\
 & & & \\
 & l & \in & \mathit{Locations} \\
\text{Region names} & r & \in & \mathit{Rnames} \\
\text{Region placeholders} & \rho & ::= & r \mid \bullet \\
\text{Stack names} & s & \in & \mathit{Snames} \\
\text{Stack placeholders} & \sigma & ::= & s \mid \circ \\
 & & & \\
\text{Computation types} & \tau & ::= & \mathsf{int} \mid \mathsf{bool} \mid \tau_1 \to \tau_2 \mid \tau_1 \times \cdots \times \tau_n \mid \alpha \mid \forall\alpha.\tau \mid \\
 & & & \mathsf{RGN}\ \tau_r\ \tau_a \mid \mathsf{RGNVar}\ \tau_r\ \tau_a \mid \mathsf{RGNHandle}\ \tau_r \mid \\
 & & & \sigma\sharp\rho \\
 & & & \\
\text{Computation terms} & e & ::= & i \mid e_1 \oplus e_2 \mid e_1 \otimes e_3 \mid \\
 & & & \mathsf{tt} \mid \mathsf{ff} \mid \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f \mid \\
 & & & x \mid \lambda x : \tau.e \mid e_1\ e_2 \mid \\
 & & & (e_1, \ldots, e_n) \mid \mathsf{sel}_i\ e \mid \\
 & & & \Lambda\alpha.e \mid e\ [\tau] \mid \\
 & & & \mathsf{let}\ x = e_1\ \mathsf{in}\ e_2 \mid \\
 & & & \mathsf{runRGN}\ [\tau_a]\ v \mid \kappa^v \mid \\
 & & & \langle l \rangle_{\sigma\sharp\rho} \mid \mathsf{handle}(\sigma\sharp\rho) \\
\text{Computation commands} & \kappa^v & ::= & \underline{\mathsf{returnRGN}}\ [\tau_r]\ [\tau_a]\ v \mid \underline{\mathsf{thenRGN}}\ [\tau_r]\ [\tau_a]\ [\tau_b]\ v_1\ v_2 \mid \\
 & & & \underline{\mathsf{letRGN}}\ [\tau_r]\ [\tau_a]\ v \mid \underline{\mathsf{witnessRGN}}\ \sigma\sharp\rho_1\ \sigma\sharp\rho_2\ [\tau_a]\ v \mid \\
 & & & \underline{\mathsf{newRGNVar}}\ [\tau_r]\ [\tau_a]\ v_1\ v_2 \mid \underline{\mathsf{readRGNVar}}\ [\tau_r]\ [\tau_a]\ v \mid \\
\text{Computation values} & v & ::= & i \mid \mathsf{tt} \mid \mathsf{ff} \mid x \mid \lambda x : \tau.e \mid (v_1, \ldots, v_n) \mid \Lambda\alpha.e \mid \kappa^v \mid \\
 & & & \langle l \rangle_{\sigma\sharp\rho} \mid \mathsf{handle}(\sigma\sharp\rho) \\
 & & & \\
\text{Regions} & R & ::= & \{l_1 \mapsto v_1, \ldots, l_n \mapsto v_n\} \\
\text{Region stacks / Stacks} & S & ::= & \cdot \mid S, r \mapsto R \quad \text{(ordered domain)} \\
\text{Stack towers / Towers} & T & ::= & \cdot \mid T, s \mapsto S \quad \text{(ordered domain)}
\end{array}
$$

$$
\begin{array}{rcl}
T \sqsupseteq_{=sr} T' & \equiv & \mathit{dom}(T) = \mathit{dom}(T') \wedge \forall s \in \mathit{dom}(T').T(s) \sqsupseteq_{sr} T'(s) \\
T \sqsupseteq_{\sqsupseteq sr} T' & \equiv & \mathit{dom}(T) \sqsupseteq \mathit{dom}(T') \wedge \forall s \in \mathit{dom}(T').T(s) \sqsupseteq_{sr} T'(s) \\
S \sqsupseteq_{=r} S' & \equiv & \mathit{dom}(S) = \mathit{dom}(S') \wedge \forall r \in \mathit{dom}(S').S(r) \sqsupseteq_{r} S'(r) \\
S \sqsupseteq_{\sqsupseteq r} S' & \equiv & \mathit{dom}(S) \sqsupseteq \mathit{dom}(S') \wedge \forall r \in \mathit{dom}(S').S(r) \sqsupseteq_{r} S'(r) \\
R \sqsupseteq_{=} R' & \equiv & \mathit{dom}(R) = \mathit{dom}(R') \wedge \forall l \in \mathit{dom}(R').R(l) = R'(l) \\
R \sqsupseteq_{\sqsupseteq} R' & \equiv & \mathit{dom}(R) \sqsupseteq \mathit{dom}(R') \wedge \forall l \in \mathit{dom}(R').R(l) = R'(l)
\end{array}
$$

Figure 21: Computation syntax of $\mathsf{F}^{\mathsf{RGN}}$

The computation syntax adds two new expression forms. The expression $\langle l \rangle_{\sigma\sharp\rho}$ is the runtime representation of a $\mathsf{RGN}\ \sigma\sharp\rho\ \tau_a$; that is, it is the pointer associated with a region allocated value. Likewise, the expression $\mathsf{handle}(\sigma\sharp\rho)$ is the runtime representation of a region handle ($\mathsf{RGNHandle}\ \sigma\sharp\rho$).

The computation syntax also adds a new command form. The command $\underline{\mathsf{witnessRGN}}\ \sigma\sharp\rho_1\ \sigma\sharp\rho_2\ [\tau_a]\ v$ casts a computation from the type $\mathsf{RGN}\ \sigma\sharp\rho_1\ \tau_a$ to the type $\mathsf{RGN}\ \sigma\sharp\rho_2\ \tau_a$. Operationally, such a command is the identify function, so long as the cast is valid. The static semantics of the next section ensure that all such casts in a well-typed program are valid.

Thus far, we have talked about region allocated data without discussing where such data is stored. We formalize the syntactic class of storable values. Storable values are associated with locations in regions R; regions are ordered into stacks S; finally, stacks are ordered into towers T. Again, towers are a technical device that serve to distinguish nested runRGN computations from one another. Intuitively, executing a runRGN computation adds a new stack to the top of the tower (which is deallocated upon finishing the computation), while executing a $\underline{\mathsf{letRGN}}$ command adds a new region to the top of the topmost stack (which is deallocated upon finishing the command). These intuitions are formalized in the operational semantics of the next section.

Finally, we introduce relations of the form $\supseteq_{tsr}$, $\supseteq_{sr}$, and $\supseteq_r$, which describe a family of extensions of towers, stacks, and regions, respectively. These relations are only needed for the type-soundness proof, but we find it convenient to state their definitions along with the definitions of towers, stacks, and regions. Note that we consider towers and stacks to have ordered domains. Hence, $dom(T) = dom(T')$ indicates that T and T' have equal ordered domains, while $dom(T) \supseteq dom(T')$ indicates that the ordered domain of $dom(T')$ is a prefix of the ordered domain of $dom(T)$. Similar comments apply to $dom(S) = dom(S')$ and $dom(S) \supseteq dom(S')$.

3.3 Dynamic semantics of $\mathsf{F}^{\mathsf{RGN}}$

Two mutually inductive judgements (Figure 22) define the dynamic semantics. We state without proof that the dynamic semantics is (almost) deterministic; it is syntax-directed, but fresh stack names, region names, and locations are chosen non-deterministicly.

The judgement $T; e \hookrightarrow v$ asserts that evaluating the closed expression e in tower T results in a value v. Likewise, the judgement $T, s \mapsto S; \kappa^v \hookrightarrow S'; v$ asserts that evaluating the closed monadic command κ^v in non-empty tower whose top stack is S results in a new top stack S' and a value v.

The rules for $T; e \hookrightarrow v$ for expression forms other than runRGN are completely standard. The tower T is passed unchanged to sub-evaluations. The rule for $\mathsf{runRGN}\ [\tau_a]\ v$ runs a monadic computation. The rule executes in the following manner. First, a fresh stack name s is chosen. Next, the argument v is applied to the region index $s\sharp r$ and the region handle $\mathsf{handle}(s\sharp r)$ and evaluated in the extended tower $T, s \mapsto \cdot, r \mapsto \{\}$ (that is, the tower T extended with a stack consisting of a single empty region (bound to r) bound to s) to a monadic command $\kappa^{v\prime}$. This command is evaluated under the extended tower to a modified region and a value v'''. The modified region is discarded, while occurrences of $s\sharp r$ are replaced by $\circ\sharp\bullet$ in v''', because the stack and region have been conceptually deallocated and are no longer accessible.

The rules for $T, s \mapsto S; \kappa^v \hookrightarrow S'; v$ peform monadic operations that side-effect the top stack. The monadic unit and bind operations are standard; in particular, note the manner in which the rule for $\underline{\mathsf{thenRGN}}$ threads the modified top stack through the computation.

The rule for $\underline{\mathsf{letRGN}}\ [s\sharp r_1]\ \tau_a\ v$ executes in much the same way as the rule for runRGN. First, a fresh region name r_2 is chosen. Next, the argument v is applied to the region index $s\sharp r_2$, a witness function, and the region handle $\mathsf{handle}(s\sharp r_2)$ and evaluated under an extended tower that adds an empty region bound to r_2 to the top of the top stack. This evaluation yields a monadic command $\kappa^{v\prime}$, which is also evaluated under the extended tower to a modified top stack and value v'''. The modified top region is discarded, while occurrences of $s\sharp r_2$ are replaced by $s\sharp\bullet$ in the modified top stack and in v''', because the region has been deallocated and is no longer accessible.

The rule for $\underline{\mathsf{witnessRGN}}$ permits a monadic computation to occur when the region names r_1 and r_2 appear in order in the top stack.

$\boxed{T; e \hookrightarrow v}$

$$\frac{}{T; i \hookrightarrow i} \qquad \frac{T; e_1 \hookrightarrow v_1 \quad v_1 \equiv i_1 \quad T; e_2 \hookrightarrow v_2 \quad v_2 \equiv i_2 \quad i = i_1 \oplus i_2}{T; e_1 \oplus e_2 \hookrightarrow i} \qquad \frac{T; e_1 \hookrightarrow v_1 \quad v_1 \equiv i_1 \quad T; e_2 \hookrightarrow v_2 \quad v_2 \equiv i_2 \quad b = i_1 \otimes i_2}{T; e_1 \otimes e_2 \hookrightarrow b}$$

$$\frac{}{T; \mathsf{tt} \hookrightarrow \mathsf{tt}} \qquad \frac{}{T; \mathsf{ff} \hookrightarrow \mathsf{ff}}$$

$$\frac{T; e_b \hookrightarrow v_b \quad v_b \equiv \mathsf{tt} \quad T; e_t \hookrightarrow v}{T; \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f \hookrightarrow v} \qquad \frac{T; e_b \hookrightarrow v_b \quad v_b \equiv \mathsf{ff} \quad T; e_f \hookrightarrow v}{T; \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f \hookrightarrow v} \qquad \frac{}{T; \lambda x : \tau . e \hookrightarrow \lambda x : \tau . e}$$

$$\frac{T; e_1 \hookrightarrow v_1 \quad v_1 \equiv \lambda x : \tau_1 . e' \quad T; e_2 \hookrightarrow v_2 \quad T; e'[v_2/x] \hookrightarrow v_3}{T; e_1\ e_2 \hookrightarrow v_3} \qquad \frac{T; e_1 \hookrightarrow v_1 \quad \ldots \quad T; e_n \hookrightarrow v_n}{T; (e_1, \ldots, e_n) \hookrightarrow (v_1, \ldots, v_n)} \qquad \frac{T; e \hookrightarrow v \quad v \equiv (v_1, \ldots, v_n) \quad 1 \leq i \leq n}{T; \mathsf{sel}_i\ e \hookrightarrow v_i}$$

$$\frac{}{T; \Lambda\alpha . e \hookrightarrow \Lambda\alpha . e} \qquad \frac{T; e \hookrightarrow v \quad v \equiv \Lambda\alpha . e' \quad T; e'[\tau/\alpha] \hookrightarrow v'}{T; e\ [\tau] \hookrightarrow v'} \qquad \frac{T; e_1 \hookrightarrow v_1 \quad T; e_2[v_1/x] \hookrightarrow v_2}{T; \mathsf{let}\ x = e_1\ \mathsf{in}\ e_2 \hookrightarrow v_2}$$

$$\frac{s \notin dom(T) \quad T, s \mapsto \cdot, r \mapsto \{\}; v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) \hookrightarrow v' \quad v' \equiv \kappa^{v'} \quad T, s \mapsto \cdot, r \mapsto \{\}; \kappa^{v'} \hookrightarrow_\kappa S''; v''' \quad S'' \equiv \cdot, r \mapsto R'''}{T; \mathsf{runRGN}\ [\tau_a]\ v \hookrightarrow v'''[s\sharp \bullet / s\sharp r][\circ/s]} \qquad \frac{}{T; \kappa^v \hookrightarrow \kappa^v} \qquad \frac{}{T; \langle l \rangle_{\sigma\sharp\rho} \hookrightarrow \langle l \rangle_{\sigma\sharp\rho}}$$

$$\frac{}{T; \mathsf{handle}(\sigma\sharp\rho) \hookrightarrow \mathsf{handle}(\sigma\sharp\rho)}$$

$\boxed{T, s \mapsto S; \kappa^v \hookrightarrow_\kappa S'; v}$

$$\frac{\tau_r \equiv s\sharp r}{T, s \mapsto S; \underline{\mathsf{returnRGN}}\ [\tau_r]\ [\tau_a]\ v \hookrightarrow_\kappa S; v} \qquad \frac{\tau_r \equiv s\sharp r \quad v_1 \equiv {\kappa^v}_1 \quad T, s \mapsto S; {\kappa^v}_1 \hookrightarrow_\kappa S'; v_1' \quad T, s \mapsto S'; v_2\ v_1' \hookrightarrow v'' \quad v'' \equiv \kappa^{v''} \quad T, s \mapsto S'; \kappa^{v''} \hookrightarrow_\kappa S'''; v'''}{T, s \mapsto S; \underline{\mathsf{thenRGN}}\ [\tau_r]\ [\tau_a]\ [\tau_b]\ v_1\ v_2 \hookrightarrow_\kappa S'''; v'''}$$

$$\frac{\tau_r \equiv s\sharp r_1 \quad r_1 \in dom(S) \quad r_2 \notin dom(S) \quad T, s \mapsto S, r_2 \mapsto \{\}; v\ [s\sharp r_2]\ (\Lambda\beta . \lambda k : \mathsf{RGN}\ s\sharp r_1\ \beta . \underline{\mathsf{witnessRGN}}\ s\sharp r_1\ s\sharp r_2\ [\beta]\ k)\ \mathsf{handle}(s\sharp r_2) \hookrightarrow v' \quad v' \equiv \kappa^{v'} \quad T, s \mapsto S, r_2 \mapsto \{\}; \kappa^{v'} \hookrightarrow_\kappa S''; v''' \quad S'' \equiv S''', r_2 \mapsto R_2'''}{T, s \mapsto S; \underline{\mathsf{letRGN}}\ [\tau_r]\ [\tau_a]\ v \hookrightarrow_\kappa S'''[s\sharp \bullet / s\sharp r_2]; v'''[s\sharp \bullet / s\sharp r_2]}$$

$$\frac{\sigma\sharp\rho_1 \equiv s\sharp r_1 \quad \sigma\sharp\rho_2 \equiv s\sharp r_2 \quad v \equiv \kappa^v \quad S \equiv S_1, r_1 \mapsto R_1, S_2, r_2 \mapsto R_2, S_3 \quad T, s \mapsto S; \kappa^v \hookrightarrow_\kappa S'; v'}{T, s \mapsto S; \underline{\mathsf{witnessRGN}}\ \sigma\sharp\rho_1\ \sigma\sharp\rho_2\ [\tau_a]\ v \hookrightarrow_\kappa S'; v'} \qquad \frac{\tau_r \equiv s\sharp r \quad v_1 \equiv \mathsf{handle}(s\sharp r) \quad r \in dom(S) \quad l \notin dom(S(r))}{T, s \mapsto S; \underline{\mathsf{newRGNVar}}\ [\tau_r]\ [\tau_a]\ v_1\ v_2 \hookrightarrow_\kappa S\{(r, l) \mapsto v_2\}; \langle l \rangle_{s\sharp r}}$$

$$\frac{\tau_r \equiv s\sharp r \quad v \equiv \langle l \rangle_{s\sharp r} \quad r \in dom(S) \quad l \in dom(S(r)) \quad v' = S(r, l)}{T, s \mapsto S; \underline{\mathsf{readRGNVar}}\ [\tau_r]\ [\tau_a]\ v \hookrightarrow_\kappa S; v'}$$

Figure 22: Dynamic semantics of $\mathsf{F}^{\mathsf{RGN}}$

The rules for newRGNVar and readRGNVar respectively allocate and read region allocated data. The rule for newRGNVar requires a region handle for a region in the top stack, chooses a fresh location in the region, and returns the top stack with the value stored at the freshly chosen location and the location. The rule for readRGNVar requires a location into a region in the top stack, and returns the value stored in the location.

It is important to note that the execution of a monadic command is predicated upon the command's region index corresponding to a live region in the top stack. While it will be possible to have commands that reference deallocated stacks and regions, it will not be possible to execute them. Furthermore, the restriction to the top stack corresponds to the fact that while runRGN computations can be nested, the inner computation must complete before executing a command in the outer computation. The type system of the next section ensures that these invariants are preserved during the execution of well-typed programs.

3.3.1 Natural Transition Semantics of $\mathsf{F}^{\mathsf{RGN}}$

While the large-step semantics presented thus far suffices to describe the complete execution of a program, it cannot describe non-terminating executions or failed executions. To do so, we adopt a *natural transition semantics* [27, 24], which provides a notion of attempted or partial execution. The key idea is to model program execution as a sequence of *partial derivation trees*, which may or may not converge to a complete derivation.

Before defining partial derivation trees, we distinguish between *complete judgements* ($T; e \hookrightarrow v$ and $T, s \mapsto S; \kappa^v \hookrightarrow_\kappa S'; v$, introduced in the large-step semantics) and *pending judgments*, which are judgements of the form $T; e \hookrightarrow ?$ or $T, s \mapsto S; \kappa^v \hookrightarrow_\kappa ?$ and represent expressions and commands that need to be evaluated.

A *partial derivation tree* is an inductively defined structure given by the following grammer:

$$
\begin{array}{lcll}
\text{Predicates} & P & & \\
\text{Complete derivations} & \mathfrak{J} & ::= & [T; e \hookrightarrow v] \mid [T, s \mapsto S; \kappa^v \hookrightarrow_\kappa S'; v] \mid [P] \\
\text{Partial derivation trees} & \mathfrak{D} & ::= & \mathfrak{J} \mid \\
 & & & [T; e \hookrightarrow ?]([J_1], \ldots, [J_{k-1}], \mathfrak{D}_k)\ ^\dagger \mid \\
 & & & [T, s \mapsto S; \kappa^v \hookrightarrow_\kappa ?]([J_1], \ldots, [J_{k-1}], \mathfrak{D}_k)\ ^\ddagger \mid
\end{array}
$$

where

† There is an instance of an evaluation rule with the form

$$\frac{J_1 \quad \cdots \quad J_n}{T; e \hookrightarrow v}$$

where $n \geq k$ and

- if $J_k \equiv T_k; e_k \hookrightarrow v_k$, then $\mathfrak{D}_k = [T_k; e_k \hookrightarrow v_k]$ or $\mathfrak{D}_k = [T_k; e_k \hookrightarrow ?]$.
- if $J_k \equiv T_k, s_k \mapsto S_k; \kappa_k^v \hookrightarrow_\kappa S_k'; v_k$, then $\mathfrak{D}_k = [T_k, s_k \mapsto S_k; \kappa_k^v \hookrightarrow_\kappa S_k'; v_k]$ or $\mathfrak{D}_k = [T_k, s_k \mapsto S_k; \kappa_k^v \hookrightarrow_\kappa ?]$.
- if $J_k \equiv P_k$, then $\mathfrak{D}_k = [P_k]$.

‡ There is an instance of an evaluation rule with the form

$$\frac{J_1 \quad \cdots \quad J_n}{T, s \mapsto S; \kappa^v \hookrightarrow_\kappa S'; v}$$

where $n \geq k$ and

- if $J_k \equiv T_k; e_k \hookrightarrow v_k$, then $\mathfrak{D}_k = [T_k; e_k \hookrightarrow v_k]$ or $\mathfrak{D}_k = [T_k; e_k \hookrightarrow ?]$.
- if $J_k \equiv T_k, s_k \mapsto S_k; \kappa_k^v \hookrightarrow_\kappa S_k'; v_k$, then $\mathfrak{D}_k = [T_k, s_k \mapsto S_k; \kappa_k^v \hookrightarrow_\kappa S_k'; v_k]$ or $\mathfrak{D}_k = [T_k, s_k \mapsto S_k; \kappa_k^v \hookrightarrow_\kappa ?]$.

– if $J_k \equiv P_k$, then $\mathfrak{D}_k = [P_k]$.

Note that the definition of a partial derivation tree requires that a node labeled with a pending judgement must have children that are "compatible" with the corresponding complete judgement. Furthermore, each node of a partial derivation tree can have at most one pending judgement amongst its children; the pending judgment must be the rightmost child and the parent node must also be a pending judgement.

Figure 23 gives (a representative sample of) the rules for the natural transition semantics. The rules are derived systematically from the judgements of Figure 22. In addition, note the two "congruence" rules. Finally, it should be clear that each transition moves a partial derivation tree "closer" to a complete judgement.

Let $\longrightarrow^*$ be the reflexive, transitive closure of the $\longrightarrow$ relation.

The natural transition semantics enjoys soundness and completeness properties demonstrating that it accurately model the large-step semantics in the case of terminating computations.

Lemma 20

If $\mathfrak{D}$ is a partial derivation and $\mathfrak{D} \longrightarrow \mathfrak{D}'$,
then $\mathfrak{D}'$ is a partial derivation.

Proof. Straightforward. □

Lemma 21 (NTS Soundness)

(1) If $[T; e \hookrightarrow ?]() \longrightarrow^* \mathfrak{D}'$ and $\mathfrak{D}'$ contains no pending judgements,
then $\mathfrak{D}'$ is a complete derivation for a judgement of the form $T; e \hookrightarrow v$.

(2) If $[T, s \mapsto S; \kappa^v \hookrightarrow_\kappa ?]() \longrightarrow^* \mathfrak{D}'_\kappa$ and $\mathfrak{D}'_\kappa$ contains no pending judgements,
then $\mathfrak{D}'_\kappa$ is a complete derivation for a judgement of the form $T, s \mapsto S; \kappa^v \hookrightarrow_\kappa S'; v$.

Proof.

(1) By Lemma 20, $\mathfrak{D}'$ is a partial derivation. Since $\mathfrak{D}'$ contains no pending judgements, $\mathfrak{D}'$ must be a complete derivation of the label on its root node. Furthemore, this label must be of the form $T; e \hookrightarrow v$, since the transitions that change a node's label change $[T; e \hookrightarrow ?](\mathfrak{J}_1, \ldots, \mathfrak{J}_n)$ to $[T; e \hookrightarrow v]$ and $[T, s \mapsto S; \kappa^v \hookrightarrow_\kappa ?](\mathfrak{J}_1, \ldots, \mathfrak{J}_n)$ to $[T, s \mapsto S; \kappa^v \hookrightarrow_\kappa S'; v]$.

(2) By Lemma 20, $\mathfrak{D}'_\kappa$ is a partial derivation. Since $\mathfrak{D}'_\kappa$ contains no pending judgements, $\mathfrak{D}'_\kappa$ must be a complete derivation of the label on its root node. Furthemore, this label must be of the form $T; e \hookrightarrow v$, since the transitions that change a node's label change $[T; e \hookrightarrow ?](\mathfrak{J}_1, \ldots, \mathfrak{J}_n)$ to $[T; e \hookrightarrow v]$ and $[T, s \mapsto S; \kappa^v \hookrightarrow_\kappa ?](\mathfrak{J}_1, \ldots, \mathfrak{J}_n)$ to $[T, s \mapsto S; \kappa^v \hookrightarrow_\kappa S'; v]$.

□

Lemma 22 (NTS Completeness)

(1) If $T; e \hookrightarrow v$ and $\mathfrak{D}$ is a complete derivation for $T; e \hookrightarrow v$,
then $[T; e \hookrightarrow ?]() \longrightarrow^* \mathfrak{D}$.

(2) If $T, s \mapsto S; \kappa^v \hookrightarrow_\kappa S'; v$ and $\mathfrak{D}_\kappa$ is a complete derivation for $T, s \mapsto S; \kappa^v \hookrightarrow_\kappa S'; v$,
then $[T, s \mapsto S; \kappa^v \hookrightarrow_\kappa ?]() \longrightarrow^* \mathfrak{D}_\kappa$.

Proof. By simultaneous induction on the derivations of $T; e \hookrightarrow v$ and $T, s \mapsto S; \kappa^v \hookrightarrow_\kappa S'; v$. □

For each tower T and expression e, we define an execution of e in T as a sequence

$$[T; e \hookrightarrow ?]() \longrightarrow \mathfrak{D}_1 \longrightarrow \mathfrak{D}_2 \longrightarrow \cdots$$

Thus, an execution has three possibilities:

$\boxed{\mathfrak{D} \longrightarrow \mathfrak{D}'}$

$$[T; \lambda x : \tau.e \hookrightarrow ?]() \longrightarrow \left[\overline{T; \lambda x : \tau.e \hookrightarrow \lambda x : \tau.e}\right] \qquad [T; e_1\ e_2 \hookrightarrow ?]() \longrightarrow [T; e_1\ e_2 \hookrightarrow ?]([T; e_1 \hookrightarrow ?]())$$

$$\frac{v_1 \equiv \lambda x : \tau_1.e_1'}{[T; e_1\ e_2 \hookrightarrow ?]([T; e_1 \hookrightarrow v_1]) \longrightarrow [T; e_1\ e_2 \hookrightarrow ?]([T; e_1 \hookrightarrow v_1], [v_1 \equiv \lambda x : \tau_1.e_1'])} \qquad [T; e_1\ e_2 \hookrightarrow ?]([T; e_1 \hookrightarrow v_1], [v_1 \equiv \lambda x : \tau_1.e_1']) \longrightarrow [T; e_1\ e_2 \hookrightarrow ?]([T; e_1 \hookrightarrow v_1], [v_1 \equiv \lambda x : \tau_1.e_1'], [T; e_2 \hookrightarrow ?]())$$

$$[T; e_1\ e_2 \hookrightarrow ?]([T; e_1 \hookrightarrow v_1], [v_1 \equiv \lambda x : \tau_1.e_1'], [T; e_2 \hookrightarrow v_2]) \longrightarrow [T; e_1\ e_2 \hookrightarrow ?]([T; e_1 \hookrightarrow v_1], [v_1 \equiv \lambda x : \tau_1.e_1'], [T; e_2 \hookrightarrow v_2], [T; e_1'[v_2/x] \hookrightarrow ?]())$$

$$[T; e_1\ e_2 \hookrightarrow ?]\begin{pmatrix}[T; e_1 \hookrightarrow v_1], [v_1 \equiv \lambda x : \tau_1.e_1'],\\ [T; e_2 \hookrightarrow v_2], [T; e_1'[v_2/x] \hookrightarrow v_3]\end{pmatrix} \longrightarrow \left[\frac{T; e_1 \hookrightarrow v_1 \qquad v_1 \equiv \lambda x : \tau_1.e' \qquad T; e_2 \hookrightarrow v_2 \qquad T; e'[v_2/x] \hookrightarrow v_3}{T; e_1\ e_2 \hookrightarrow v_3}\right]$$

$$\frac{s \notin dom(T)}{[T; \mathsf{runRGN}\ [\tau_a]\ v \hookrightarrow ?]() \longrightarrow [T; \mathsf{runRGN}\ [\tau_a]\ v \hookrightarrow ?]([s \notin T])} \qquad [T; \mathsf{runRGN}\ [\tau_a]\ v \hookrightarrow ?]([s \notin T]) \longrightarrow [T; \mathsf{runRGN}\ [\tau_a]\ v \hookrightarrow ?]([s \notin T], [T, s \mapsto \cdot, r \mapsto \{\}; v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) \hookrightarrow ?]())$$

$$\frac{v' \equiv \kappa^{v'}}{[T; \mathsf{runRGN}\ [\tau_a]\ v \hookrightarrow ?]([s \notin T], [T, s \mapsto \cdot, r \mapsto \{\}; v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) \hookrightarrow v'], [v' \equiv \kappa^{v'}]) \longrightarrow [T; \mathsf{runRGN}\ [\tau_a]\ v \hookrightarrow ?]([s \notin T], [T, s \mapsto \cdot, r \mapsto \{\}; v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) \hookrightarrow v'], [v' \equiv \kappa^{v'}])}$$

$$[T; \mathsf{runRGN}\ [\tau_a]\ v \hookrightarrow ?]\begin{pmatrix}[s \notin T],\\ [T, s \mapsto \cdot, r \mapsto \{\}; v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) \hookrightarrow v'], [v' \equiv \kappa^{v'}]\end{pmatrix} \longrightarrow [T; \mathsf{runRGN}\ [\tau_a]\ v \hookrightarrow ?]\begin{pmatrix}[s \notin T],\\ [T, s \mapsto \cdot, r \mapsto \{\}; v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) \hookrightarrow v'], [v' \equiv \kappa^{v'}],\\ [T, s \mapsto \cdot, r \mapsto \{\}; \kappa^{v'} \hookrightarrow_\kappa ?]()\end{pmatrix}$$

$$\frac{S'' \equiv \cdot; r \mapsto R'''}{[T; \mathsf{runRGN}\ [\tau_a]\ v \hookrightarrow ?]\begin{pmatrix}[s \notin T],\\ [T, s \mapsto \cdot, r \mapsto \{\}; v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) \hookrightarrow v'], [v' \equiv \kappa^{v'}],\\ [T, s \mapsto \cdot, r \mapsto \{\}; \kappa^{v'} \hookrightarrow_\kappa S''; v''']\end{pmatrix} \longrightarrow [T; \mathsf{runRGN}\ [\tau_a]\ v \hookrightarrow ?]\begin{pmatrix}[s \notin T],\\ [T, s \mapsto \cdot, r \mapsto \{\}; v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) \hookrightarrow v'], [v' \equiv \kappa^{v'}],\\ [T, s \mapsto \cdot, r \mapsto \{\}; \kappa^{v'} \hookrightarrow_\kappa S''; v'''], [S'' \equiv \cdot; r \mapsto R''']\end{pmatrix}}$$

$$[T; \mathsf{runRGN}\ [\tau_a]\ v \hookrightarrow ?]\begin{pmatrix}[s \notin T],\\ [T, s \mapsto \cdot, r \mapsto \{\}; v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) \hookrightarrow v'], [v' \equiv \kappa^{v'}],\\ [T, s \mapsto \cdot, r \mapsto \{\}; \kappa^{v'} \hookrightarrow_\kappa S''; v'''], [S'' \equiv \cdot; r \mapsto R''']\end{pmatrix} \longrightarrow \left[\frac{s \notin dom(T) \qquad T, s \mapsto \cdot, r \mapsto \{\}; v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) \hookrightarrow v' \qquad v' \equiv \kappa^{v'} \qquad T, s \mapsto \cdot, r \mapsto \{\}; \kappa^{v'} \hookrightarrow_\kappa S''; v''' \qquad S'' \equiv \cdot, r \mapsto R'''}{T; \mathsf{runRGN}\ [\tau_a]\ v \hookrightarrow v'''[s\sharp \bullet / s\sharp r][\circ/s]}\right]$$

$$\frac{\mathfrak{D} \longrightarrow \mathfrak{D}'}{[T; e \hookrightarrow ?](\mathfrak{J}_1, \ldots, \mathfrak{J}_k, \mathfrak{D}) \longrightarrow [T; e \hookrightarrow ?](\mathfrak{J}_1, \ldots, \mathfrak{J}_k, \mathfrak{D}')} \qquad \frac{\mathfrak{D} \longrightarrow \mathfrak{D}'}{[T, s \mapsto S; \kappa^v \hookrightarrow_\kappa ?](\mathfrak{J}_1, \ldots, \mathfrak{J}_k, \mathfrak{D}) \longrightarrow [T, s \mapsto S; \kappa^v \hookrightarrow_\kappa ?](\mathfrak{J}_1, \ldots, \mathfrak{J}_k, \mathfrak{D}')}$$

Figure 23: Natural transition semantics of $\mathsf{F}^{\mathsf{RGN}}$

(1) Suppose that for all $\mathfrak{D}_n$ such that $[T; e \hookrightarrow ?]() \longrightarrow^* \mathfrak{D}_n$, there exists $\mathfrak{D}_{n+1}$ such that $\mathfrak{D}_n \longrightarrow \mathfrak{D}_{n+1}$. Then, we say that e in T diverges.

(2) Suppose that there exists $\mathfrak{D}_n$ such that $[T; e \hookrightarrow ?]() \longrightarrow^* \mathfrak{D}_n$, such that there does not exist $\mathfrak{D}_{n+1}$ such that $\mathfrak{D}_n \longrightarrow \mathfrak{D}_{n+1}$.

   (a) Suppose $\mathfrak{D}_n$ contains no pending judgements. By Lemma 21, $\mathfrak{D}_n \equiv [T; e \hookrightarrow v]$. Then, we say that e in T terminates with the value v.

   (b) Suppose $\mathfrak{D}_n$ contains pending judgements. Then, we say that e in T gets stuck.

By inspection of the rules in Figure 23, it is clear that the stuck partial derivation trees correspond to trees in which predicates cannot be satisfied; all other transitions are unrestricted. Predicates like $v \equiv \lambda x : \tau.e$ and $v \equiv \kappa^v$ are traditional type errors, where expressions evaluate to values of the wrong form. Predicates like $r \in dom(S)$ also correspond to type errors, where towers have the wrong form. The static semantics given in the next section and the definitions given in Section 3.5.3 ensure that stuck partial derivation trees are not well-typed.

3.4 Static Semantics of $\mathsf{F}^{\mathsf{RGN}}$

Well-typed programs obey several invariants, which are enforced with typing judgements. In addition to the traditional "type-checking" judgements for expressions, we have judgements that enforce the consistency of towers, and various well-formedness judgements that serve as a technical convenience.

3.4.1 Definitions

Figure 24 present additional definitions for syntactic classes that appear in the static semantics. Type contexts Δ are ordered lists of type variables and value contexts Γ are ordered lists of variables and types. Tower, stack, and region types mimic towers, stacks, and regions, recording the type of the value stored at each location. Tower, stack, and region domains are a technical device that records the locations in scope. Because proving the well-formedness of tower types requires proving the well-formedness of types, which requires verifying that stack and region names are in scope, one cannot easily have tower types serve the dual purpose of recording locations in scope.

We introduce the (suggestive) abbreviation $\tau_r' \preceq \tau_r$ for a function that coerces any computation taking place in the region indexed by τ_r' into a computation taking place in the region indexed by τ_r. We call such functions *witnesses* and explain their role in more detail below.

We overload the relations of the form $\supseteq_{tsr}$, $\supseteq_{sr}$, and $\supseteq_r$ to describe extensions of tower, stack, and region domains and types. The same conventions described above for ordered domains apply. Finally, we define restriction operators, which return a prefix of tower domains and types.

3.4.2 Expressions

Figures 25, 26, and 28 present the typing rules for the judgement $\Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e : \tau$, which asserts that under type context Δ, value context Γ, and tower type $\mathcal{T}$ with tower domain $\overline{\mathcal{T}}$, the expression e has type τ.

The rules for constants, arithmetic and boolean operations, function abstraction and application, tuple introduction and elimination, and type abstraction and instantiation are all completely standard. As expected in a monadic language, each command expression is given the monadic type $\mathsf{RGN}\ \tau_r\ \tau_a$ for appropriate region index and return type. The typing rules for returnRGN and thenRGN correspond to the standard typing rules for monadic unit and bind operations. The typing rules for newRGNVar and readRGNVar are straight-forward.

As in the Single Effect Calculus, the key judgements are those relating to the creation of new regions. We first examine the typing rule for the runRGN expression:

$$\frac{\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau_a \qquad \Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v : \forall \alpha.\mathsf{RGNHandle}\ \alpha \to \mathsf{RGN}\ \alpha\ \tau_a}{\Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \mathsf{runRGN}\ [\tau_a]\ v : \tau_a}$$

$$
\begin{array}{lrcl}
\text{Type contexts} & \Delta & ::= & \cdot \mid \Delta, \alpha \\
\text{Value contexts} & \Gamma & ::= & \cdot \mid \Gamma, x : \tau \\
\text{Region domains} & \overline{\mathcal{R}} & ::= & \{l_1, \ldots, l_n\} \\
\text{Region types} & \mathcal{R} & ::= & \{l_1 \mapsto \tau_1, \ldots, l_n \mapsto \tau_n\} \\
\text{Region stack domains / Stack domains} & \overline{\mathcal{S}} & ::= & \cdot \mid \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \quad \text{(ordered domain)} \\
\text{Region stack types / Stack types} & \mathcal{S} & ::= & \cdot \mid \mathcal{S}, r \mapsto \mathcal{R} \quad \text{(ordered domain)} \\
\text{Stack tower domains / Tower domains} & \overline{\mathcal{T}} & ::= & \cdot \mid \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \quad \text{(ordered domain)} \\
\text{Stack tower types / Tower types} & \mathcal{T} & ::= & \cdot \mid \mathcal{T}, s \mapsto \mathcal{S} \quad \text{(ordered domain)}
\end{array}
$$

$$
\tau_{r'} \preceq \tau_r \quad \equiv \quad \forall \beta.\mathsf{RGN}\ \tau_{r'}\ \beta \to \mathsf{RGN}\ \tau_r\ \beta
$$

$$
\begin{array}{rcl}
\overline{\mathcal{T}} \supseteq_{=sr} \overline{\mathcal{T}}' & \equiv & dom(\overline{\mathcal{T}}) = dom(\overline{\mathcal{T}}') \wedge \forall s \in dom(\overline{\mathcal{T}}').\overline{\mathcal{T}}(s) \supseteq_{sr} \overline{\mathcal{T}}'(s) \\
\overline{\mathcal{T}} \supseteq_{\supseteq sr} \overline{\mathcal{T}}' & \equiv & dom(\overline{\mathcal{T}}) \supseteq dom(\overline{\mathcal{T}}') \wedge \forall s \in dom(\overline{\mathcal{T}}').\overline{\mathcal{T}}(s) \supseteq_{sr} \overline{\mathcal{T}}'(s) \\
\overline{\mathcal{S}} \supseteq_{=r} \overline{\mathcal{S}}' & \equiv & dom(\overline{\mathcal{S}}) = dom(\overline{\mathcal{S}}') \wedge \forall r \in dom(\overline{\mathcal{S}}').\overline{\mathcal{S}}(r) \supseteq_{r} \overline{\mathcal{S}}'(r) \\
\overline{\mathcal{S}} \supseteq_{\supseteq r} \overline{\mathcal{S}}' & \equiv & dom(\overline{\mathcal{S}}) \supseteq dom(\overline{\mathcal{S}}') \wedge \forall r \in dom(\overline{\mathcal{S}}').\overline{\mathcal{S}}(r) \supseteq_{r} \overline{\mathcal{S}}'(r) \\
\overline{\mathcal{R}} \supseteq_{=} \overline{\mathcal{R}}' & \equiv & dom(\overline{\mathcal{R}}) = dom(\overline{\mathcal{R}}') \\
\overline{\mathcal{R}} \supseteq_{\supseteq} \overline{\mathcal{R}}' & \equiv & dom(\overline{\mathcal{R}}) \supseteq dom(\overline{\mathcal{R}}')
\end{array}
$$

$$
\overline{\mathcal{T}}|_s \quad \equiv \quad \overline{\mathcal{T}}', s \mapsto \overline{\mathcal{S}}' \quad \text{such that } \overline{\mathcal{T}} \equiv \overline{\mathcal{T}}', s \mapsto \overline{\mathcal{S}}', \overline{\mathcal{T}}''
$$

$$
\begin{array}{rcl}
\mathcal{T} \supseteq_{=sr} \mathcal{T}' & \equiv & dom(\mathcal{T}) = dom(\mathcal{T}') \wedge \forall s \in dom(\mathcal{T}').\mathcal{T}(s) \supseteq_{sr} \mathcal{T}'(s) \\
\mathcal{T} \supseteq_{\supseteq sr} \mathcal{T}' & \equiv & dom(\mathcal{T}) \supseteq dom(\mathcal{T}') \wedge \forall s \in dom(\mathcal{T}').\mathcal{T}(s) \supseteq_{sr} \mathcal{T}'(s) \\
\mathcal{S} \supseteq_{=r} \mathcal{S}' & \equiv & dom(\mathcal{S}) = dom(\mathcal{S}') \wedge \forall r \in dom(\mathcal{S}').\mathcal{S}(r) \supseteq_{r} \mathcal{S}'(r) \\
\mathcal{S} \supseteq_{\supseteq r} \mathcal{S}' & \equiv & dom(\mathcal{S}) \supseteq dom(\mathcal{S}') \wedge \forall r \in dom(\mathcal{S}').\mathcal{S}(r) \supseteq_{r} \mathcal{S}'(r) \\
\mathcal{R} \supseteq_{=} \mathcal{R}' & \equiv & dom(\mathcal{R}) = dom(\mathcal{R}') \wedge \forall l \in dom(\mathcal{R}').\mathcal{R}(l) = \mathcal{R}'(l) \\
\mathcal{R} \supseteq_{\supseteq} \mathcal{R}' & \equiv & dom(\mathcal{R}) \supseteq dom(\mathcal{R}') \wedge \forall l \in dom(\mathcal{R}').\mathcal{R}(l) = \mathcal{R}'(l)
\end{array}
$$

$$
\mathcal{T}|_s \quad \equiv \quad \mathcal{T}', s \mapsto \mathcal{S}' \quad \text{such that } \mathcal{T} \equiv \mathcal{T}', s \mapsto \mathcal{S}', \mathcal{T}''
$$

Figure 24: Static semantics of $\mathsf{F}^{\mathsf{RGN}}$ (definitions)

$\boxed{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e : \tau}$

$$\frac{\vdash_{\text{ctxt}} \Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}}}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} i : \mathsf{int}} \qquad \frac{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_1 : \mathsf{int} \quad \Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_2 : \mathsf{int}}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_1 \oplus e_2 : \mathsf{int}} \qquad \frac{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_1 : \mathsf{int} \quad \Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_2 : \mathsf{int}}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_1 \otimes e_2 : \mathsf{bool}}$$

$$\frac{\vdash_{\text{vctxt}} \Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}}}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \mathsf{tt} : \mathsf{bool}} \qquad \frac{\vdash_{\text{vctxt}} \Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}}}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \mathsf{ff} : \mathsf{bool}} \qquad \frac{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_b : \mathsf{bool} \quad \Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_t : \tau \quad \Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_f : \tau}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f : \tau}$$

$$\frac{\vdash_{\text{ctxt}} \Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \in dom(\Gamma)}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} x : \tau} \qquad \frac{\Delta;\Gamma, x : \tau_1;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e : \tau_2}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \lambda x : \tau_1.e : \tau_1 \to \tau_2} \qquad \frac{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_1 : \tau_1 \to \tau_2 \quad \Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_2 : \tau_1}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_1\ e_2 : \tau_2}$$

$$\frac{\vdash_{\text{ctxt}} \Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \quad \Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_i : \tau_i \ {}^{i \in 1 \ldots n}}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} (e_1, \ldots, e_n) : \tau_1 \times \cdots \times \tau_n} \qquad \frac{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e : \tau_1 \times \cdots \times \tau_n \quad 1 \le i \le n}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \mathsf{sel}_i\ e : \tau_i} \qquad \frac{\vdash_{\text{ctxt}} \Delta;\Gamma;\mathcal{S} : \overline{\mathcal{S}} \quad \Delta, \alpha;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e : \tau}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \Lambda\alpha.e : \forall\alpha.\tau}$$

$$\frac{\Delta;\overline{\mathcal{T}} \vdash_{\text{type}} \tau \quad \Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e : \forall\alpha.\tau'}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e\ [\tau] : \tau'[\tau/\alpha]} \qquad \frac{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_1 : \tau_1 \quad \Delta;\Gamma, x : \tau_1;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_2 : \tau_2}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \mathsf{let}\ x = e_1\ \mathsf{in}\ e_2 : \tau_2}$$

$$\frac{\Delta;\overline{\mathcal{T}} \vdash_{\text{type}} \tau_a \qquad \Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v : \forall\alpha.\mathsf{RGNHandle}\ \alpha \to \mathsf{RGN}\ \alpha\ \tau_a}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \mathsf{runRGN}\ [\tau_a]\ v : \tau_a}$$

Figure 25: Static semantics of $\mathsf{F}^{\mathsf{RGN}}$ (expressions)

$\boxed{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e : \tau}$

$$\frac{\Delta;\overline{\mathcal{T}} \vdash_{\text{type}} \tau_r \quad \Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v : \tau_a}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \underline{\mathsf{returnRGN}}\ [\tau_r]\ [\tau_a]\ v : \mathsf{RGN}\ \tau_r\ \tau_a} \qquad \frac{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_1 : \mathsf{RGN}\ \tau_r\ \tau_a \quad \Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_2 : \tau_a \to \mathsf{RGN}\ \tau_r\ \tau_b}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \underline{\mathsf{thenRGN}}\ [\tau_r]\ [\tau_a]\ [\tau_b]\ v_1\ v_2 : \mathsf{RGN}\ \tau_r\ \tau_b}$$

$$\frac{\Delta;\overline{\mathcal{T}} \vdash_{\text{type}} \tau_r \qquad \Delta;\overline{\mathcal{T}} \vdash_{\text{type}} \tau_a \quad \Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v : \forall\alpha.\tau_r \preceq \alpha \to \mathsf{RGNHandle}\ \alpha \to \mathsf{RGN}\ \alpha\ \tau_a}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \underline{\mathsf{letRGN}}\ [\tau_r]\ [\tau_a]\ v : \mathsf{RGN}\ \tau_r\ \tau_a}$$

$$\frac{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v : \mathsf{RGN}\ \sigma\sharp\rho_1\ \tau_a \qquad \overline{\mathcal{T}} \vdash_{\text{cast}} \sigma\sharp\rho_1 \rightsquigarrow \sigma\sharp\rho_2}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \underline{\mathsf{witnessRGN}}\ \sigma\sharp\rho_1\ \sigma\sharp\rho_2\ [\tau_a]\ v : \mathsf{RGN}\ \sigma\sharp\rho_2\ \tau_a}$$

$$\frac{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v_1 : \mathsf{RGNHandle}\ \tau_r \qquad \Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v_2 : \tau_a}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \underline{\mathsf{newRGNVar}}\ [\tau_r]\ [\tau_a]\ v_1\ v_2 : \mathsf{RGN}\ \tau_r\ (\mathsf{RGNVar}\ \tau_r\ \tau_a)}$$

$$\frac{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v : \mathsf{RGNVar}\ \tau_r\ \tau_a}{\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \underline{\mathsf{readRGNVar}}\ [\tau_r]\ [\tau_a]\ v : \mathsf{RGN}\ \tau_r\ \tau_a}$$

Figure 26: Static semantics of $\mathsf{F}^{\mathsf{RGN}}$ (commands)

$\boxed{\overline{\mathcal{T}} \vdash_{\text{cast}} \sigma \sharp \rho \rightsquigarrow \sigma \sharp \rho'}$

$$\frac{\vdash_{\text{tdom}} \overline{\mathcal{T}}}{\overline{\mathcal{T}} \vdash_{\text{cast}} \circ \sharp \bullet \rightsquigarrow \circ \sharp \bullet} \qquad \frac{\vdash_{\text{tdom}} \overline{\mathcal{T}} \qquad s \in dom(\overline{\mathcal{T}})}{\overline{\mathcal{T}} \vdash_{\text{cast}} s \sharp \bullet \rightsquigarrow s \sharp \bullet} \qquad \frac{\vdash_{\text{tdom}} \overline{\mathcal{T}} \qquad s \in dom(\overline{\mathcal{T}}) \qquad \overline{\mathcal{T}}(s) \equiv \overline{\mathcal{S}}_1, r_1 \mapsto \overline{\mathcal{R}}_1, \overline{\mathcal{S}}_2}{\overline{\mathcal{T}} \vdash_{\text{cast}} s \sharp r_1 \rightsquigarrow s \sharp \bullet} \qquad \frac{\vdash_{\text{tdom}} \overline{\mathcal{T}} \qquad s \in dom(\overline{\mathcal{T}}) \qquad \overline{\mathcal{T}}(s) \equiv \overline{\mathcal{S}}_1, r_1 \mapsto \overline{\mathcal{R}}_1, \overline{\mathcal{S}}_2, r_2 \mapsto \overline{\mathcal{R}}_2, \overline{\mathcal{S}}_3}{\overline{\mathcal{T}} \vdash_{\text{cast}} s \sharp r_1 \rightsquigarrow s \sharp r_2}$$

Figure 27: Static semantics of $\mathsf{F}^{\mathsf{RGN}}$ (casts)

As stated above, the argument to runRGN should describe a region computation. In fact, we require v to be a polymorphic function that yields a region computation after being applied to a region handle. Recall that we can consider a value of type RGN τ_r τ_a as a region-transformer – that is, it accepts a region (indexed by the type τ_r), performs some operations (such as allocating into the region), and returns a value and the modified region. The effect of universally quantifying the region index in the type of v is to require v to make no assumptions about the input region (e.g., the existence of pre-allocated values). Furthermore, all operations that manipulate a region are "infected" with the region index: when combining operations, the rule for thenRGN requires the region index type to be the same; locations allocated and read using newRGNVar and readRGNVar require the region index of the RGNVar to be the same as the computation in which the operation occurs. Thus, if a region computation RGN τ_r τ_a were to return a value that depended upon the region indexed by τ_r, then τ_r would appear in the type τ_a. Since the type τ_a appears outside the scope of the type variable α in the typing rule for runRGN, it follows that α cannot appear in the type τ_a. Therefore, it must be the case that the value returned by the computation described by v does not depend upon the region index which will instantiate α. Taken together, these facts ensure that an arbitrary new region can be supplied to the computation and that the value returned will not leak any means of accessing the region or values allocated within it; hence, the region can be destroyed at the end of the computation. Finally, because we require region handles for allocating within regions, we provide the region handle for the newly created region as the argument to a function that yields the computation we wish to execute.

The typing rule for letRGN is very similar:

$$\frac{\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau_r \qquad \Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau_a \qquad \Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v : \forall \alpha . \tau_r \preceq \alpha \rightarrow \mathsf{RGNHandle}\ \alpha \rightarrow \mathsf{RGN}\ \alpha\ \tau_a}{\Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \underline{\mathsf{letRGN}}\ [\tau_r]\ [\tau_a]\ v : \mathsf{RGN}\ \tau_r\ \tau_a}$$

Ignoring for the moment the argument of type $\tau_r \preceq \alpha$, we see that exactly the same argument as above applies. In particular, the computation makes no assumptions about the newly created region, nor can the region be leaked through the returned value of type τ_a. What, then, is the role of the witness argument? The answer lies in the fact that we do not really intend the execution to take place in an arbitrary region. Instead, we expect the newly allocated region to be related to previously allocated regions according to a stack discipline (just as in region calculi). Hence, the notion of "execution taking place in a region" is somewhat inaccurate; instead, we have executions taking place in a stack of regions. The region index in a type RGN τ_r τ_a indicates a particular member of the region stack; in practice, it often coincides with the most recently allocated region. Thus, any computation taking place in a stack of regions where τ_r' is a member (i.e., a RGN τ_r' τ_a term) is also a computation taking place in a stack of regions where τ_r is a member (i.e., a RGN τ_r τ_a term) whenever τ_r' outlives τ_r. A function of type $\tau_r' \preceq \tau_r$ witnesses this coercion. This explains the role of the witness argument – it is provided to the computation taking place in the inner region in order to coerce computations (such as allocating a new value in the outer region) from the outer region to the inner region. Operationally, such a witness function acts as the identity function.

The typing rule for witnessRGN formalizes this outlives argument; a witnessRGN term is well-typed whenever $\sigma \sharp \rho_1$ can be cast to $\sigma \sharp \rho_2$. The judgement $\overline{\mathcal{T}} \vdash_{\text{cast}} \sigma \sharp \rho_1 \rightsquigarrow \sigma \sharp \rho_2$ (Figure 27) verifies the casts witnessed by witnessRGN terms. Note that the judgment $\overline{\mathcal{T}} \vdash_{\text{cast}} s \sharp r_1 \rightsquigarrow s \sharp r_2$ enforces the requirement that

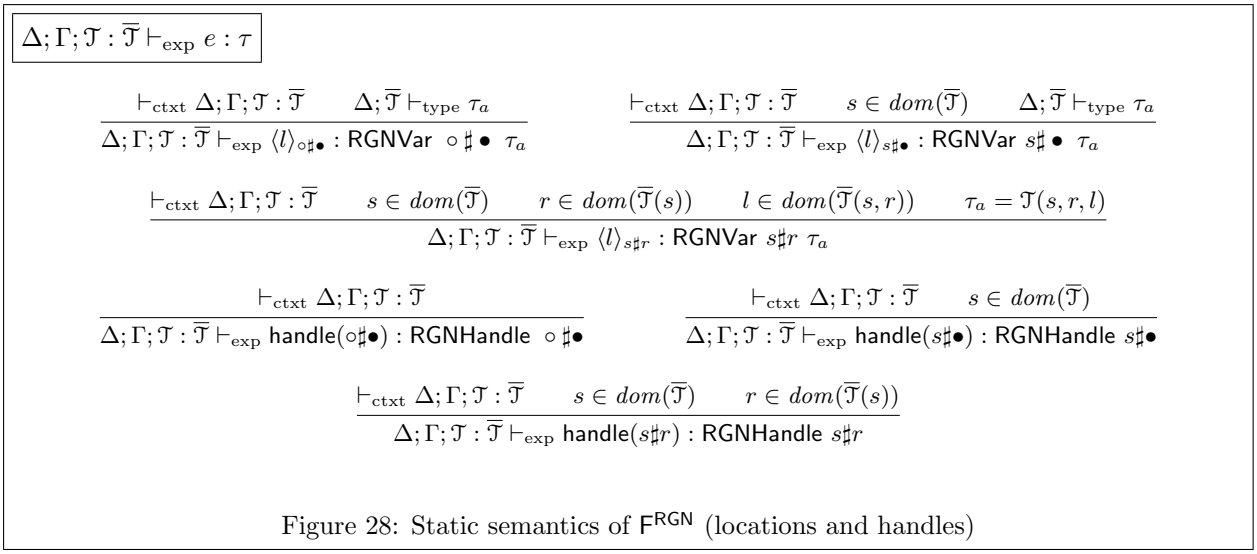


Figure 28: Static semantics of $\mathsf{F}^{\mathsf{RGN}}$ (locations and handles)

r_1 outlives r_2 in the stack s. The other $\vdash_{\text{cast}}$ judgements allow casts to deallocated regions, which can be introduced when deallocating a region at the end of a runRGN or letRGN computation.

Figure 28 type-check location and handle expressions. The judgements ensure that stack and region names that appear in locations and handles are in scope; furthermore, a location in a live stack and region points to a value with the type assigned by the tower type.

3.4.3 Towers

Figure 29 present typing rules that enforce the well-formedness and consistency of towers. The judgements $\vdash_{\text{tdom}}$, $\vdash_{\text{sdom}}$, and $\vdash_{\text{rdom}}$ simply require that tower, stack, and region domains contain distinct stack names, region names, and locations. The judgement $\vdash_{\text{ttype}} \mathcal{T} : \overline{\mathcal{T}}$ asserts that tower type $\mathcal{T}$ is well-formed with tower domain $\overline{\mathcal{T}}$. In particular, the judgement asserts that $\mathcal{T}$ has the domain specified by $\overline{\mathcal{T}}$ and each type in the range of $\mathcal{T}$ is well-formed. Note the use of the restriction operator; this ensures that types "lower" in the tower cannot reference stack and region names that appear "higher" in the tower. This corresponds to the fact that while runRGN computations can be nested, the inner computation must complete before executing a command in the outer computation. Hence, while an inner computation may have references to the outer computation, there can be no references from the outer computation to the inner computation. Finally, the judgement $\vdash_{\text{tower}} T : \mathcal{T} : \overline{\mathcal{T}}$ asserts that the tower T is well-formed with tower type $\mathcal{T}$ and tower domain $\overline{\mathcal{T}}$. Like the judgement $\vdash_{\text{ttype}}$, it asserts that T has the domain specified by $\overline{\mathcal{T}}$ and each stored value in the range of T has the type specified by $\mathcal{T}$. Again, restriction operators are used to assert that storables "lower" in the tower cannot contain references to storables "higher" in the tower.

3.4.4 Technical details

Figures 30 and 31 contain additional judgements for ensuring that types τ, type contexts Δ, and value contexts Γ are well-formed. Because types may contain stack and region names, the judgements $\vdash_{\text{type}}$ and $\vdash_{\text{vctxt}}$ require a tower domain $\overline{\mathcal{T}}$.

3.4.5 Surface programs

It is useful to note that the static semantics can be greatly simplified for the surface syntax presented above. Tower types and tower domains are purely technical devices that are used to prove type soundness. In the static semantics, they simply collect the names of stacks and regions in scope and assign types to locations. Note that in every rule, tower types and tower domains are passed unmodified to sub-judgements.

$\boxed{\vdash_{\text{tdom}} \overline{\mathcal{T}}}$

$$\frac{}{\vdash_{\text{tdom}} \cdot} \qquad \frac{\vdash_{\text{tdom}} \overline{\mathcal{T}} \qquad s \notin dom(\overline{\mathcal{T}}) \qquad \vdash_{\text{sdom}} \overline{\mathcal{S}}}{\vdash_{\text{tdom}} \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}}$$

$\boxed{\vdash_{\text{sdom}} \overline{\mathcal{S}}}$

$$\frac{}{\vdash_{\text{sdom}} \cdot} \qquad \frac{\vdash_{\text{sdom}} \overline{\mathcal{S}} \qquad r \notin dom(\overline{\mathcal{S}}) \qquad \vdash_{\text{rdom}} \overline{\mathcal{R}}}{\vdash_{\text{sdom}} \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}}$$

$\boxed{\vdash_{\text{rdom}} \overline{\mathcal{R}}}$

$$\frac{i \neq j \Rightarrow l_i \neq l_j \ ^{i \in 1 \ldots n, j \in 1 \ldots n}}{\vdash_{\text{rdom}} \{l_1, \ldots, l_n\}}$$

$\boxed{\vdash_{\text{ttype}} \mathcal{T} : \overline{\mathcal{T}}}$

$$\frac{\begin{array}{c}\vdash_{\text{tdom}} \overline{\mathcal{T}} \\ dom(\overline{\mathcal{T}}) = dom(\mathcal{T}) \\ \forall s \in dom(\overline{\mathcal{T}}).\ dom(\overline{\mathcal{T}}(s)) = dom(\mathcal{T}(s)) \\ \forall s \in dom(\overline{\mathcal{T}}). \forall r \in dom(\overline{\mathcal{T}}(s))\ dom(\overline{\mathcal{T}}(s,r)) = dom(\mathcal{T}(s,r)) \\ \forall s \in dom(\overline{\mathcal{T}}). \forall r \in dom(\overline{\mathcal{T}}(s)). \forall l \in dom(\overline{\mathcal{T}}(s,r)).\ \cdot; \overline{\mathcal{T}}|_s \vdash_{\text{type}} \mathcal{T}(s,r,l)\end{array}}{\vdash_{\text{ttype}} \mathcal{T} : \overline{\mathcal{T}}}$$

$\boxed{\vdash_{\text{tower}} T : \mathcal{T} : \overline{\mathcal{T}}}$

$$\frac{\begin{array}{c}\vdash_{\text{tdom}} \overline{\mathcal{T}} \\ \vdash_{\text{ttype}} \mathcal{T} : \overline{\mathcal{T}} \\ dom(\overline{\mathcal{T}}) = dom(\mathcal{T}) = dom(T) \\ \forall s \in dom(\overline{\mathcal{T}}).\ dom(\overline{\mathcal{T}}(s)) = dom(\mathcal{T}(s)) = dom(T(s)) \\ \forall s \in dom(\overline{\mathcal{T}}). \forall r \in dom(\overline{\mathcal{T}}(s))\ dom(\overline{\mathcal{T}}(s,r)) = dom(\mathcal{T}(s,r)) = dom(T(s,r)) \\ \forall s \in dom(\overline{\mathcal{T}}). \forall r \in dom(\overline{\mathcal{T}}(s)). \forall l \in dom(\overline{\mathcal{T}}(s,r)).\ \cdot; \cdot; \mathcal{T}|_s : \overline{\mathcal{T}}|_s \vdash_{\text{exp}} T(s,r,l) : \mathcal{T}(s,r,l)\end{array}}{\vdash_{\text{tower}} T : \mathcal{T} : \overline{\mathcal{T}}}$$

Figure 29: Static semantics of $\mathsf{F}^{\mathsf{RGN}}$ (towers, stacks, and regions)

$\boxed{\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau}$

$$\frac{\vdash_{\text{tdom}} \overline{\mathcal{T}}_{\text{tctxt}}\Delta}{\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \mathsf{int}} \qquad \frac{\vdash_{\text{tdom}} \overline{\mathcal{T}}_{\text{tctxt}}\Delta}{\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \mathsf{bool}} \qquad \frac{\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau_1 \qquad \Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau_2}{\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau_1 \to \tau_2} \qquad \frac{\vdash_{\text{tdom}} \overline{\mathcal{T}}_{\text{tctxt}}\Delta \quad \Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau_i \;^{i \in 1 \ldots n}}{\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau_1 \times \cdots \times \tau_n}$$

$$\frac{\vdash_{\text{tdom}} \overline{\mathcal{T}}_{\text{tctxt}}\Delta \quad \alpha \in \mathit{dom}(\Delta)}{\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \alpha} \qquad \frac{\Delta, \alpha; \overline{\mathcal{T}} \vdash_{\text{type}} \tau}{\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \forall \alpha.\tau} \qquad \frac{\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau_r \qquad \Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau_a}{\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \mathsf{RGN}\ \tau_r\ \tau_a} \qquad \frac{\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau_r \qquad \Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau_a}{\Delta; \overline{\mathcal{T}} \vdash \mathsf{RGNVar}\ \tau_r\ \tau_a}$$

$$\frac{\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau_r}{\Delta; \overline{\mathcal{T}} \vdash \mathsf{RGNHandle}\ \tau_r} \qquad \frac{\vdash_{\text{tdom}} \overline{\mathcal{T}} \qquad \vdash_{\text{tctxt}} \Delta}{\Delta; \overline{\mathcal{T}} \vdash \circ\sharp\bullet} \qquad \frac{\vdash_{\text{tdom}} \overline{\mathcal{T}} \qquad \vdash_{\text{tctxt}} \Delta \quad s \in \mathit{dom}(\overline{\mathcal{T}})}{\Delta; \overline{\mathcal{T}} \vdash s\sharp\bullet} \qquad \frac{\vdash_{\text{tdom}} \overline{\mathcal{T}} \qquad \vdash_{\text{tctxt}} \Delta \quad s \in \mathit{dom}(\overline{\mathcal{T}}) \qquad r \in \mathit{dom}(\overline{\mathcal{T}}(s))}{\Delta; \overline{\mathcal{T}} \vdash s\sharp r}$$

Figure 30: Static semantics of $\mathsf{F}^{\mathsf{RGN}}$ (types)

$\boxed{\vdash_{\text{tctxt}} \Delta}$

$$\frac{}{\vdash_{\text{tctxt}} \cdot} \qquad \frac{\vdash_{\text{tctxt}} \Delta \qquad \alpha \notin \mathit{dom}(\Delta)}{\vdash_{\text{tctxt}} \Delta, \alpha}$$

$\boxed{\Delta; \overline{\mathcal{T}} \vdash_{\text{vctxt}} \Gamma}$

$$\frac{\vdash_{\text{tdom}} \overline{\mathcal{T}}_{\text{tctxt}}\Delta}{\Delta; \overline{\mathcal{T}} \vdash_{\text{vctxt}} \cdot} \qquad \frac{\Delta; \overline{\mathcal{T}} \vdash_{\text{vctxt}} \Gamma \qquad x \notin \mathit{dom}(\Gamma) \qquad \Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau}{\Delta; \overline{\mathcal{T}} \vdash_{\text{vctxt}} \Gamma, x : \tau}$$

$\boxed{\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}}}$

$$\frac{\vdash_{\text{ttype}} \mathcal{T} : \overline{\mathcal{T}}; \overline{\mathcal{T}} \vdash_{\text{vctxt}} \Gamma}{\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}}}$$

Figure 31: Static semantics of $\mathsf{F}^{\mathsf{RGN}}$ (contexts)

Since surface programs do not admit syntax for naming stacks and regions, we can type any closed, surface expression with the judement $\cdot;\cdot;\cdot : \cdot \vdash_{\text{exp}} e : \tau$. Pushing these empty tower types and tower domains through the rules leads to the following simplifications:

$\Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e : \tau$	$\Longrightarrow$	$\Delta;\Gamma \vdash_{\text{exp}} e : \tau$
$\Delta;\mathcal{T} \vdash_{\text{type}} \tau$	$\Longrightarrow$	$\Delta \vdash_{\text{type}} \tau$
$\Delta;\mathcal{T} \vdash_{\text{vctxt}} \Gamma$	$\Longrightarrow$	$\Delta \vdash_{\text{vctxt}} \Gamma$
$\vdash_{\text{ctxt}} \Delta;\Gamma;\mathcal{T} : \overline{\mathcal{T}}$	$\Longrightarrow$	$\vdash_{\text{ctxt}} \Delta;\Gamma$

Hence, we recover a type system equivalent to that of System F, which is sufficient for type-checking surface programs.

Further simplifications can be made by interpreting the monadic commands as constants with polymorphic types. For example, the typing judgements for each of the monadic commands are equivalent to the following typings:

$$
\begin{array}{rcl}
\mathsf{runRGN} & :: & \forall\alpha.(\forall\beta.\mathsf{RGNHandle}\ \beta \to \mathsf{RGN}\ \beta\ \alpha) \to \alpha \\
\underline{\mathsf{returnRGN}} & :: & \forall\alpha.\forall\beta.\alpha \to \mathsf{RGN}\ \beta\ \alpha \\
\underline{\mathsf{thenRGN}} & :: & \forall\alpha.\forall\alpha'.\forall\beta.\mathsf{RGN}\ \beta\ \alpha \to (\alpha \to \mathsf{RGN}\ \beta\ \alpha') \to \mathsf{RGN}\ \beta\ \alpha' \\
\underline{\mathsf{letRGN}} & :: & \forall\alpha.\forall\beta.(\forall\gamma.\beta \preceq \gamma \to \mathsf{RGNHandle}\ \gamma \to \mathsf{RGN}\ \gamma\ \alpha) \to \mathsf{RGN}\ \beta\ \alpha \\
\underline{\mathsf{newRGNVar}} & :: & \forall\alpha.\forall\beta.\mathsf{RGNHandle}\ \beta \to \alpha \to \mathsf{RGN}\ \beta\ (\mathsf{RGNVar}\ \beta\ \alpha) \\
\underline{\mathsf{readRGNVar}} & :: & \forall\alpha.\forall\beta.\mathsf{RGNVar}\ \beta\ \alpha \to \mathsf{RGN}\ \beta\ \alpha
\end{array}
$$

Treating the monadic commands as syntactic forms simplifies the proofs, as there is no need to consider partially applied forms.

3.5 Type Soundness of $\mathsf{F}^{\mathsf{RGN}}$

In this section, we prove type soundness. The lemmas are presented in "bottom-up" order, where all relevant lemmas are presented (and proved) before being used. The lemmas follow the conventional structure of a syntactic type-soundness arguments. However, we first give a "top-down" overview of the argument.

We wish to prove that a well-typed, closed initial program either succeeds (returning a value of the correct type) or runs forever. A preservation theorem and a progress theorem make this theorem an easy corollary.

The Preservation Theorem states that the terminating computation of a well-typed expression yields a value of the same type. Because the dynamic semantics are defined by two mutually inductive judgements, the Preservation Theorem also states that the terminating computation of a well-typed command yields a well-typed extension of the top stack and a value of the same type. The various substitution lemmas for dead stacks and regions are required to prove the cases where stacks and regions are deallocated.

The Progress Theorem states that a partially evaluated expression can always move forward towards complete evaluation. Progress Theorems are notoriously difficult in a large-step semantics. Our approach adopts a *natural transition semantics*, introduced in Section 3.3.1. The Progess Theorem states that any well-typed partial derivation that contains a pending judgement can transition to another well-typed partial derivation. As usual, the proof of the Progess Theorem depends on a Canonical Forms Lemma, which describes the forms of values of particular types.

The various Well-Formedness Lemmas are useful technical facts that alleviate the need to assume contexts are well-formed.

3.5.1 Lemmas

Lemma 23 (Well-Formedness (from $\vdash_{\text{type}}$))

If $\Delta;\overline{\mathcal{T}} \vdash_{\text{type}} \tau$,
then $\vdash_{\text{tdom}} \overline{\mathcal{T}}$ *and* $\vdash_{\text{tctxt}} \Delta$.

Proof. By induction on the derivation $\Delta;\overline{\mathcal{T}} \vdash_{\text{type}} \tau$. □

Lemma 24 (Well-Formedness (from $\vdash_{\text{vctxt}}$))

If $\Delta; \overline{\mathcal{T}} \vdash_{\text{vctxt}} \Gamma$,
then $\vdash_{\text{tdom}} \overline{\mathcal{T}}$ *and* $\vdash_{\text{tctxt}} \Delta$.

Proof. By induction on the derivation $\Delta; \overline{\mathcal{T}} \vdash_{\text{vctxt}} \Gamma$. □

Lemma 25 (Well-Formedness ($\vdash_{\text{ctxt}}$))

If $\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}}$,
then $\vdash_{\text{tdom}} \overline{\mathcal{T}}$, $\vdash_{\text{ttype}} \mathcal{T} : \overline{\mathcal{T}}$, $\vdash_{\text{tctxt}} \Delta$, *and* $\Delta; \mathcal{T} \vdash_{\text{vctxt}} \Gamma$.

Proof. By inspection of the derivation $\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}}$. □

Lemma 26 (Well-Formedness (from $\vdash_{\text{cast}}$))

If $\overline{\mathcal{T}} \vdash_{\text{cast}} \sigma\sharp\rho_1 \rightsquigarrow \sigma\sharp\rho_2$,
then $\vdash_{\text{tdom}} \overline{\mathcal{T}}$, $\cdot; \overline{\mathcal{T}} \vdash_{\text{type}} \sigma\sharp\rho_1$, *and* $\cdot; \overline{\mathcal{T}} \vdash_{\text{type}} \sigma\sharp\rho_2$.

Proof. By inspection of the derivation $\overline{\mathcal{T}} \vdash_{\text{cast}} \sigma\sharp\rho_1 \rightsquigarrow \sigma\sharp\rho_2$. □

Lemma 27 (Substitution (types in $\vdash_{\text{type}}$))

If $\Delta, \alpha, \Delta'; \overline{\mathcal{T}} \vdash_{\text{type}} \tau'$ *and* $\Delta, \Delta'; \overline{\mathcal{T}} \vdash_{\text{type}} \tau$,
then $\Delta, \Delta'; \overline{\mathcal{T}} \vdash_{\text{type}} \tau'[\tau/\alpha]$.

Proof. By induction on the derivation $\Delta, \alpha, \Delta'; \overline{\mathcal{T}} \vdash_{\text{type}} \tau'$. □

Lemma 28 (Type Context Weakening ($\vdash_{\text{type}}$))

If $\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau$, $\vdash_{\text{tctxt}} \Delta, \Delta'$, *and* $dom(\Delta') \cap \text{btv}(\tau) = \emptyset$,
then $\Delta, \Delta'; \overline{\mathcal{T}} \vdash_{\text{type}} \tau$.

Proof. By induction on the derivation $\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau$. □

Lemma 29 (Well-Formedness (from $\vdash_{\text{exp}}$))

If $\Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e : \tau$,
then $\vdash_{\text{tdom}} \overline{\mathcal{T}}$, $\vdash_{\text{ttype}} \mathcal{T} : \overline{\mathcal{T}}$, $\vdash_{\text{tctxt}} \Delta$, $\Delta; \overline{\mathcal{T}} \vdash_{\text{vctxt}} \Gamma$, *and* $\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau$.

Proof. By induction on the derivation $\Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e : \tau$. □

Lemma 30 (Substitution (types in $\vdash_{\text{exp}}$))

If $\Delta, \alpha, \Delta'; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e' : \tau'$ *and* $\Delta, \Delta'; \overline{\mathcal{T}} \vdash_{\text{type}} \tau$,
then $\Delta, \Delta'; \Gamma[\tau/\alpha]; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e'[\tau/\alpha] : \tau'[\tau/\alpha]$.

Proof. By induction on the derivation $\Delta, \alpha, \Delta'; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e' : \tau'$. □

Lemma 31 (Substitution (values in $\vdash_{\text{exp}}$))

If $\Delta; \Gamma, x : \tau, \Gamma'; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e' : \tau'$ *and* $\Delta; \Gamma, \Gamma'; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v : \tau$,
then $\Delta; \Gamma, \Gamma'; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e'[v/x] : \tau'$.

Proof. By induction on the derivation $\Delta; \Gamma, x : \tau, \Gamma'; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e' : \tau'$. □

Lemma 32 (Tower Domain Weakening)

Suppose $\overline{\mathcal{T}'} \supseteq_{tsr} \overline{\mathcal{T}}$ *and* $\vdash_{\text{tdom}} \overline{\mathcal{T}'}$.

(1) If $\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau$,
then $\Delta; \overline{\mathcal{T}'} \vdash_{\text{type}} \tau$.

(2) If $\Delta; \overline{\mathcal{T}} \vdash_{\text{vctxt}} \Gamma$,
then $\Delta; \overline{\mathcal{T}'} \vdash_{\text{vctxt}} \Gamma$.

Further suppose $\mathcal{T}' \supseteq_{tsr} \mathcal{T}$ *and* $\vdash_{\text{ttype}} \mathcal{T}' : \overline{\mathcal{T}'}$.

(3) If $\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}}$,
then $\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{T}' : \overline{\mathcal{T}'}$.

(4) If $\overline{\mathcal{T}} \vdash_{\text{cast}} \sigma\sharp\rho_1 \rightsquigarrow \sigma\sharp\rho_2$,
then $\overline{\mathcal{T}'} \vdash_{\text{cast}} \sigma\sharp\rho_1 \rightsquigarrow \sigma\sharp\rho_2$.

(5) If $\Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e : \tau$,
then $\Delta; \Gamma; \mathcal{T}' : \overline{\mathcal{T}'} \vdash_{\text{exp}} e : \tau$.

Proof.

(1) By induction on the derivation $\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau$.

(2) By induction on the derivation $\Delta; \overline{\mathcal{T}} \vdash_{\text{vctxt}} \Gamma$.

(3) By inspection of the derivation $\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}}$.

(4) By inspection of the derivation $\overline{\mathcal{T}} \vdash_{\text{cast}} \sigma\sharp\rho_1 \rightsquigarrow \sigma\sharp\rho_2$.

(5) By induction on the derivation $\Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e : \tau$.

□

Lemma 33 (Substitution (•))

(1) If $\vdash_{\text{tdom}} \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}$,
then $\vdash_{\text{tdom}} \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$.

(2) If $\Delta; \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{type}} \tau$,
then $\Delta; \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{type}} \tau[s\sharp \bullet / s\sharp r]$.

(3) If $\vdash_{\text{ttype}} \mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}$,
then $\vdash_{\text{ttype}} \mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet / s\sharp r] : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$.

(4) If $\Delta; \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{vctxt}} \Gamma$,
then $\Delta; \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}[s\sharp \bullet / s\sharp r] \vdash_{\text{vctxt}} \Gamma[s\sharp \bullet / s\sharp r]$.

(5) If $\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}$,
then $\vdash_{\text{ctxt}} \Delta; \Gamma[s\sharp \bullet / s\sharp r]; \mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet / s\sharp r] : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$.

(6) If $\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{cast}} \sigma\sharp\rho_1 \rightsquigarrow \sigma\sharp\rho_2$,
then $\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{cast}} (\sigma\sharp\rho_1)[s\sharp \bullet / s\sharp r] \rightsquigarrow (\sigma\sharp\rho_2)[s\sharp \bullet / s\sharp r]$.

(7) If $\Delta; \Gamma; \mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{exp}} e : \tau$,
then $\Delta; \Gamma[s\sharp \bullet / s\sharp r]; \mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet / s\sharp r] : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} e[s\sharp \bullet / s\sharp r] : \tau[s\sharp \bullet / s\sharp r]$,

(8) If $\vdash_{\text{tower}} T, s \mapsto S, r \mapsto R : \mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}$,
then $\vdash_{\text{tower}} T, s \mapsto S[s\sharp \bullet / s\sharp r] : \mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet / s\sharp r] : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$.

Proof.

(1) By inspection of the derivation of $\vdash_{\text{tdom}} \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}$.

(2) By induction on the derivation $\Delta; \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{type}} \tau$.

(3) By inspection, the derivation of $\vdash_{\text{ttype}} \mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}$ must end with

$$\frac{\begin{array}{c}\vdash_{\text{tdom}} \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \\ dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}) = dom(\mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R}) \\ \forall s' \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}). \\ dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}})(s')) = dom((\mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R})(s')) \\ \forall s' \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}).\forall r' \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}})(s')) \\ dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}})(s', r')) = dom((\mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R})(s', r')) \\ \forall s' \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}).\forall r' \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}})(s')).\forall l' \in dom((\overline{\mathcal{T}}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R})(s', r')). \\ \cdot; (\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}})|_{s'} \vdash_{\text{type}} (\mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R})(s', r', l')\end{array}}{\vdash_{\text{ttype}} \mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}}$$

By (1), we conclude $\vdash_{\text{tdom}} \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$. We are required to show

$$\frac{\begin{array}{c}\vdash_{\text{tdom}} \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \\ dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}) = dom(\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r]) \\ \forall s' \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}). \\ dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})(s')) = dom((\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(s')) \\ \forall s' \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}).\forall r' \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}) \\ dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})(s', r')) = dom((\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(s', r')) \\ \forall s' \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}).\forall r' \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})(s')).\forall l' \in dom((\overline{\mathcal{T}}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(s', r')). \\ \cdot; (\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})|_{s'} \vdash_{\text{type}} (\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(s', r', l')\end{array}}{\vdash_{\text{ttype}} \mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r] : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}}$$

Clearly, all of the domain equalities hold. It remains to show

$$\forall s' \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}).\forall r' \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})(s')).\forall l' \in dom((\overline{\mathcal{T}}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(s', r')). \\ \cdot; (\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})|_{s'} \vdash_{\text{type}} (\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(s', r', l')$$

Note that for $s' \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})$, $s' \neq s$ and $r' \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})(s'))$,

$$(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})|_{s'} \equiv (\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}})|_{s'} \quad \text{and} \\ (\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(r', s', l') \equiv (\mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R})(s', r', l')$$

Hence, $\cdot; (\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})|_{s'} \vdash_{\text{type}} (\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(s', r', l')$. Note that for $s' = s$ and $r' \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})(s))$,

$$\cdot; (\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})|_{s} \vdash_{\text{type}} (\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(s, r', l') \equiv \\ \cdot; \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{type}} (\mathcal{S}[s\sharp \bullet /s\sharp r])(r', l') \equiv \\ \cdot; \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{type}} (\mathcal{S}(r', l'))[s\sharp \bullet /s\sharp r]$$

which follows from (1) applied to $\cdot; \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{type}} (\mathcal{S}, r \mapsto \mathcal{R})(r', l')$. Hence, $\cdot; (\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})|_{s} \vdash_{\text{type}} (\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(s, r', l')$. Thus, $\vdash_{\text{ttype}} \mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r] : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$, as required.

(4) By induction on the derivation $\Delta; \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{vctxt}} \Gamma$.

(5) By inspection of the derivation $\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}$.

(6) By inspection of the derivation $\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{cast}} \sigma\sharp\rho_1 \rightsquigarrow \sigma\sharp\rho_2$.

(7) By induction on the derivation $\Delta; \Gamma; \mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{exp}} e : \tau$.

(8) By inspection, the derivation of $\vdash_{\text{tower}} T, s \mapsto S, r \mapsto R : \mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}$ must end with

$$\frac{\begin{array}{c}\vdash_{\text{tdom}} \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \\ \vdash_{\text{ttype}} \mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \\ dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}) = dom(\mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R}) = dom(T, s \mapsto S, r \mapsto R) \\ \forall s' \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}). \\ dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}})(s')) = dom((\mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R})(s')) = dom((T, s \mapsto S, r \mapsto R)(s')) \\ \forall s' \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}).\forall r' \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}})(s')) \\ dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}})(s', r')) = dom((\mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R})(s', r')) = dom((T, s \mapsto S, r \mapsto R)(s', r')) \\ \forall s' \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}).\forall r' \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}})(s')).\forall l' \in dom((\overline{\mathcal{T}}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R})(s', r')). \\ \cdot;\cdot;(\mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R})|_{s'} : (\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}})|_{s'} \vdash_{\text{exp}} (T, s \mapsto S, r \mapsto R)(r, s, l) : (\mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R})(s', r', l')\end{array}}{\vdash_{\text{tower}} T, s \mapsto S, r \mapsto R : \mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}}$$

By (1), we conclude $\vdash_{\text{tdom}} \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$. By (3), we conclude $\vdash_{\text{ttype}} \mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r] : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$. We are required to show

$$\frac{\begin{array}{c}\vdash_{\text{tdom}} \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \\ \vdash_{\text{ttype}} \mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r] : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \\ dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}) = dom(\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r]) = dom(T, s \mapsto S[s\sharp \bullet /s\sharp r]) \\ \forall s' \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}). \\ dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})(s')) = dom((\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(s')) = dom((T, s \mapsto S[s\sharp \bullet /s\sharp r])(s')) \\ \forall s' \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}).\forall r' \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}) \\ dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})(s', r')) = dom((\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(s', r')) = dom((T, s \mapsto S[s\sharp \bullet /s\sharp r])(s', r')) \\ \forall s' \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}).\forall r' \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})(s')).\forall l' \in dom((\overline{\mathcal{T}}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(s', r')). \\ \cdot;\cdot;(\mathcal{T}, s \mapsto \mathcal{S})|_{s'} : (\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})|_{s'} \vdash_{\text{exp}} (T, s \mapsto S[s\sharp \bullet /s\sharp r])(s', r', l') : (\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(s', r', l')\end{array}}{\vdash_{\text{tower}} T, s \mapsto S[s\sharp \bullet /s\sharp r] : \mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r] : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}}$$

Clearly, all of the domain equalities hold. It remains to show

$$\begin{array}{c}\forall s' \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}).\forall r' \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})(s')).\forall l' \in dom((\overline{\mathcal{T}}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(s', r')). \\ \cdot;\cdot;(\mathcal{T}, s \mapsto \mathcal{S})|_{s'} : (\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})|_{s'} \vdash_{\text{exp}} (T, s \mapsto S[s\sharp \bullet /s\sharp r])(s', r', l') : (\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(s', r', l')\end{array}$$

Note that for $s' \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})$, $s' \neq s$ and $r' \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})(s'))$,

$$\begin{array}{c}(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})|_{s'} \equiv (\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}})|_{s'} \quad \text{and} \\ (\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])|_{s'} \equiv (\mathcal{T}, s \mapsto \mathcal{S})|_{s'} \quad \text{and} \\ (\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(r', s', l') \equiv (\mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R})(s', r', l') \quad \text{and} \\ (T, s \mapsto S[s\sharp \bullet /s\sharp r])(r', s', l') \equiv (T, s \mapsto S, r \mapsto R)(s', r', l')\end{array}$$

Hence, $\cdot;\cdot;(\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])|_{s'} : (\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})|_{s'} \vdash_{\text{exp}} (T, s \mapsto S[s\sharp \bullet /s\sharp r])(s', r', l') : (\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(s', r', l')$. Note that for $s' = s$ and $r' \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})(s))$,

$$\begin{array}{c}\cdot;\cdot;(\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])|_s : (\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})|_s \vdash_{\text{exp}} (T, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(s, r', l') : (\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(s, r', l') \equiv \\ \cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r] : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} (\mathcal{S}[s\sharp \bullet /s\sharp r])(r', l') : (\mathcal{S}[s\sharp \bullet /s\sharp r])(r', l') \equiv \\ \cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r] : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} (\mathcal{S}(r', l'))[s\sharp \bullet /s\sharp r] : (\mathcal{S}(r', l'))[s\sharp \bullet /s\sharp r]\end{array}$$

which follows from (7) applied to $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}, r \mapsto \mathcal{R} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}} \vdash_{\text{exp}} (S, r \mapsto R)(r', l') : (\mathcal{S}, r \mapsto \mathcal{R})(r', l')$. Hence, $\cdot;\cdot;(\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])|_s : (\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})|_s \vdash_{\text{exp}} (T, s \mapsto S[s\sharp \bullet /s\sharp r])(s, r', l') : (\mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r])(s, r', l')$. Thus, $\vdash_{\text{tower}} T, s \mapsto S[s\sharp \bullet /s\sharp r] : \mathcal{T}, s \mapsto \mathcal{S}[s\sharp \bullet /s\sharp r] : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$, as required.

□

Lemma 34 (Substitution ($\circ$))

(1) If $\vdash_{\text{tdom}} \mathcal{T}, s \mapsto \cdot$,
then $\vdash_{\text{tdom}} \mathcal{T}$.

(2) If $\Delta; \mathcal{T}, s \mapsto \cdot \vdash_{\text{type}} \tau$,
then $\Delta; \mathcal{T} \vdash_{\text{type}} \tau[\circ/s]$.

(3) If $\vdash_{\text{ttype}} \mathcal{T}, s \mapsto \cdot : \overline{\mathcal{T}}, s \mapsto \cdot$,
then $\vdash_{\text{ttype}} \mathcal{T} : \overline{\mathcal{T}}$.

(4) If $\Delta; \overline{\mathcal{T}}, s \mapsto \cdot \vdash_{\text{vctxt}} \Gamma$,
then $\Delta; \overline{\mathcal{T}} \vdash_{\text{vctxt}} \Gamma[\circ/s]$.

(5) If $\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{T}, s \mapsto \cdot : \overline{\mathcal{T}}, s \mapsto \cdot$,
then $\vdash_{\text{ctxt}} \Delta; \Gamma[\circ/s]; \mathcal{T} : \overline{\mathcal{T}}$.

(6) If $\overline{\mathcal{T}}, s \mapsto \cdot \vdash_{\text{cast}} \sigma\sharp\rho_1 \rightsquigarrow \sigma\sharp\rho_2$,
then $\overline{\mathcal{T}} \vdash_{\text{cast}} (\sigma\sharp\rho_1)[\circ/s] \rightsquigarrow (\sigma\sharp\rho_2)[\circ/s]$.

(7) If $\Delta; \Gamma; \mathcal{T}, s \mapsto \cdot : \overline{\mathcal{T}}, s \mapsto \cdot \vdash_{\text{exp}} e : \tau$,
then $\Delta; \Gamma[\circ/s]; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e[\circ/s] : \tau[\circ/s]$,

(8) If $\vdash_{\text{tower}} T, s \mapsto \cdot : \mathcal{T}, s \mapsto \cdot : \overline{\mathcal{T}}, s \mapsto \cdot$,
then $\vdash_{\text{tower}} T : \mathcal{T} : \overline{\mathcal{T}}$.

Proof. Similar to the proof of Lemma 33. □

Lemma 35 (Useless Substitution (α))

Suppose $\alpha \notin dom(\Delta)$ *and* $\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau'$.

(1) If $\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau$,
then $\tau[\tau'/\alpha] \equiv \tau$.

Proof. By induction on the derivation of $\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau$. □

Lemma 36 (Useless Substitution ($\bullet$))

Suppose $s \in dom(\overline{\mathcal{T}})$ *and* $r \notin dom(\overline{\mathcal{T}}(s))$.

(1) If $\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau$,
then $\tau[s\sharp \bullet / s\sharp r] \equiv \tau$.

Proof. By induction on the derivation of $\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau$. □

Lemma 37 (Useless Substitution ($\circ$))

Suppose $s \notin dom(\overline{\mathcal{T}})$.

(1) If $\Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau$,
then $\tau[\circ/s] \equiv \tau$.

Proof. Similar to the proof of Lemma 36. □

Lemma 38 (Canonical Forms)

Suppose $\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v : \tau$.
Then

- *if* $\tau \equiv \mathsf{int}$, *then* $v \equiv i$.
- *if* $\tau \equiv \mathsf{bool}$, *then* $v \equiv \mathsf{tt}$ *or* $v \equiv \mathsf{ff}$.
- *if* $\tau \equiv \tau_1 \to \tau_2$, *then* $v \equiv \lambda x : \tau_1.e$.
- *if* $\tau \equiv \tau_1 \times \cdots \times \tau_n$, *then* $v \equiv (v_1, \ldots, v_n)$.
- *if* $\tau \equiv \forall\alpha.\tau'$, *then* $v \equiv \Lambda\alpha.e'$.
- *if* $\tau \equiv \mathsf{RGN}\ \tau_r\ \tau_a$, *then* $v \equiv \kappa^v$.
- *if* $\tau \equiv \mathsf{RGNHandle}\ \sigma\sharp\rho$, *then* $v \equiv \mathsf{handle}(\sigma\sharp\rho)$.
- *if* $\tau \equiv \mathsf{RGNVar}\ \sigma\sharp\rho\ \tau_a$, *then* $v \equiv \langle l \rangle_{\sigma\sharp\rho}$.

Proof. Immediate by inspection of the typing rules. □

3.5.2 Preservation

Theorem 2 (Preservation)

(1) If

(a) $\vdash_{\text{tower}} T : \mathcal{T} : \overline{\mathcal{T}}$,
(b) $\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e : \tau$, *and*
(c) $T; e \hookrightarrow v'$,

then $\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v' : \tau$.

(2) If

(a) $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$,
(b) $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \kappa^v : \mathsf{RGN}\ s\sharp r\ \tau_a$, *and*
(c) $T, s \mapsto S; \kappa^v \hookrightarrow_\kappa S'; v'$,

then there exists $\overline{\mathcal{S}}' \supseteq_{=\supseteq} \overline{\mathcal{S}}$ *and* $\mathcal{S}' \supseteq_{=\supseteq} \mathcal{S}$ *such that* $\vdash_{\text{tower}} T, s \mapsto S' : \mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}'$ *and* $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}' \vdash_{\text{exp}} v' : \tau_a$.

Proof. Proceed by mutual induction on the derivations (1c) $T; e \hookrightarrow v'$ and (2c) $T, s \mapsto S; \kappa^v \hookrightarrow S'; v'$.

Case $\dfrac{}{T; i \hookrightarrow i}$: By inspection, the derivation of (1b) must end with

$$\dfrac{\vdash_{\text{ctxt}} \cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}}}{\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} i : \mathsf{int}}$$

We conclude $\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} i : \mathsf{int}$, as required.

Case $\dfrac{\begin{array}{cc} T; e_1 \hookrightarrow v_1 & v_1 \equiv i_1 \\ T; e_2 \hookrightarrow v_2 & v_2 \equiv i_2 \end{array} \quad i = i_1 \oplus i_2}{T; e_1 \oplus e_2 \hookrightarrow i}$: ...

Case $\dfrac{\begin{array}{cc} T; e_1 \hookrightarrow v_1 & v_1 \equiv i_1 \\ T; e_2 \hookrightarrow v_2 & v_2 \equiv i_2 \end{array} \quad b = i_1 \otimes i_2}{T; e_1 \otimes e_2 \hookrightarrow b}$: ...

Case $\dfrac{}{T; \mathsf{tt} \hookrightarrow \mathsf{tt}}$: ...

Case $\dfrac{}{T; \mathsf{ff} \hookrightarrow \mathsf{ff}}$: ...

Case $\dfrac{T; e_b \hookrightarrow v_b \qquad v_b \equiv \mathsf{tt} \qquad T; e_t \hookrightarrow v}{T; \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f \hookrightarrow v}$: ...

Case $\dfrac{T; e_b \hookrightarrow v_b \qquad v_b \equiv \mathsf{ff} \qquad T; e_f \hookrightarrow v}{T; \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f \hookrightarrow v}$: ...

Case $\dfrac{}{T; \lambda x : \tau . e \hookrightarrow \lambda x : \tau . e}$: ...

Case $\dfrac{T; e_1 \hookrightarrow v_1 \qquad v_1 \equiv \lambda x : \tau_1 . e' \qquad T; e_2 \hookrightarrow v_2 \qquad T; e'[v_2/x] \hookrightarrow v_3}{T; e_1\ e_2 \hookrightarrow v_3}$: By inspection, the derivation of (1b) must end with

$$\dfrac{\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_1 : \tau_1 \to \tau_2 \qquad \cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_2 : \tau_1}{\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_1\ e_2 : \tau_2}$$

By applying the induction hypothesis to (1a) $\vdash_{\text{tower}} T : \mathcal{T} : \overline{\mathcal{T}}$, (1b) $\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_1 : \tau_1 \to \tau_2$, and (1c) $T; e_1 \hookrightarrow \lambda x : \tau_1 . e_1'$, we conclude $\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \lambda x : \tau_1 . e_1' : \tau_1 \to \tau_2$. By inspection, this derivation must end with

$$\dfrac{\cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{type}} \tau_1 \qquad \cdot; \cdot, x : \tau_1; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_1' : \tau_2}{\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \lambda x : \tau_1 . e_1' : \tau_1 \to \tau_2}$$

By applying the induction hypothesis to (1a) $\vdash_{\text{tower}} T : \mathcal{T} : \overline{\mathcal{T}}$, (1b) $\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_2 : \tau_1$, and (1c) $T; e_2 \hookrightarrow v_2$, we conclude $\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v_2 : \tau_1$. By Lemma 31, we conclude $\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_1'[v_2/x] : \tau_2$. By applying the induction hypothesis to (1a) $\vdash_{\text{tower}} T : \mathcal{T} : \overline{\mathcal{T}}$, (1b) $\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e'[v/x] : \tau_2$, and (1c) $T; e_1'[v_2/x] \hookrightarrow v_3$, we conclude $\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v_3 : \tau_2$, as required.

Case $\dfrac{}{T; \Lambda \alpha . e \hookrightarrow \Lambda \alpha . e}$: ...

Case $\dfrac{T; e \hookrightarrow v \qquad v \equiv \Lambda \alpha . e' \qquad T; e'[\tau/\alpha] \hookrightarrow v'}{T; e\ [\tau] \hookrightarrow v'}$: By inspection, the derivation of (1b) must end with

$$\dfrac{\cdot; \overline{\mathcal{T}} \vdash_{\text{type}} \tau \qquad \cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e : \forall \alpha . \tau'}{\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e\ [\tau] : \tau'[\tau/\alpha]}$$

By applying the induction hypothesis to (1a) $\vdash_{\text{tower}} T : \mathcal{T} : \overline{\mathcal{T}}$, (1b) $\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e : \forall \alpha . \tau'$, and (1c) $T; e \hookrightarrow \Lambda \alpha . e'$, we conclude $\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \Lambda \alpha . e' : \forall \alpha . \tau'$. By inspection, this derivation must end with

$$\dfrac{\alpha \notin dom(\cdot) \qquad \cdot, \alpha; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e' : \tau'}{\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \Lambda \alpha . e' : \forall \alpha . \tau'}$$

By Lemma 30, we conclude $\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e'[\tau/\alpha] : \tau'[\tau/\alpha]$. By applying the induction hypothesis to (1a) $\vdash_{\text{tower}} T : \mathcal{T} : \overline{\mathcal{T}}$, (1b) $\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e'[\tau/\alpha] : \tau'[\tau/\alpha]$, and (1c) $T; e'[\tau/\alpha] \hookrightarrow v$, we conclude $\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v : \tau'[\tau/\alpha]$, as required.

Case $\dfrac{T; e_1 \hookrightarrow v_1 \qquad \ldots \qquad T; e_n \hookrightarrow v_n}{T; (e_1, \ldots, e_n) \hookrightarrow (v_1, \ldots, v_n)}$: ...

Case $\dfrac{T; e \hookrightarrow v \qquad v \equiv (v_1, \ldots, v_n) \qquad 1 \leq i \leq n}{T; \mathsf{sel}_i\ e \hookrightarrow v_i}$: ...

Case $\dfrac{\begin{array}{c}T; e_1 \hookrightarrow v_1 \\ T; e_2[v_1/x] \hookrightarrow v_2\end{array}}{T; \mathsf{let}\ x = e_1\ \mathsf{in}\ e_2 \hookrightarrow v_2}$: ...

Case $\dfrac{\begin{array}{c}s \notin dom(T) \\ T, s \mapsto \cdot, r \mapsto \{\}; v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) \hookrightarrow v' \qquad v' \equiv \kappa^{v'} \\ T, s \mapsto \cdot, r \mapsto \{\}; \kappa^{v'} \hookrightarrow_\kappa S''; v''' \qquad S'' \equiv \cdot, r \mapsto R'''\end{array}}{T; \mathsf{runRGN}\ [\tau_a]\ v \hookrightarrow v'''[s\sharp \bullet / s\sharp r][\circ/s]}$: By inspection, the derivation of (1b) must end with

$$\dfrac{\cdot; \overline{\mathcal{T}} \vdash_{\text{type}} \tau_a \qquad \cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v : \forall\alpha.\mathsf{RGNHandle}\ \alpha \to \mathsf{RGN}\ \alpha\ \tau_a}{\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \mathsf{runRGN}\ [\tau_a]\ v : \tau_a}$$

By inspection of the derivation $\vdash_{\text{tower}} T : \mathcal{T} : \overline{\mathcal{T}}$, we conclude $\vdash_{\text{tdom}} \overline{\mathcal{T}}$, $\vdash_{\text{ttype}} \mathcal{T} : \overline{\mathcal{T}}$, and $dom(T) = dom(\mathcal{T}) = dom(\overline{T}))$. Note that $\overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \supseteq_{\supseteq ==} \overline{\mathcal{T}}$ and $\mathcal{T}, s \mapsto \cdot, r \mapsto \{\} \supseteq_{\supseteq ==} \mathcal{T}$ and

$$\dfrac{\vdash_{\text{tdom}} \overline{\mathcal{T}} \qquad s \notin dom(\overline{\mathcal{T}}) \qquad \dfrac{\overline{\vdash_{\text{sdom}} \cdot} \qquad r \notin dom(\cdot) \qquad \overline{\vdash_{\text{rdom}} \{\}}}{\vdash_{\text{sdom}} \cdot, r \mapsto \{\}}}{\vdash_{\text{tdom}} \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\}}$$

and $\vdash_{\text{ttype}} \mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\}$ (requires appealing to Lemma 32) and $\vdash_{\text{tower}} T, s \mapsto \cdot, r \mapsto \{\} : \mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\}$ (requires appaling to Lemma 32). By Lemma 32, we conclude $\cdot; \cdot; \mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \vdash_{\text{exp}} v : \forall\alpha.\mathsf{RGNHandle}\ \to \mathsf{RGN}\ \alpha\ \tau_a$. Note that

$$\dfrac{\dfrac{\dfrac{\vdash_{\text{tdom}} \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \qquad \vdash_{\text{tctxt}} \cdot \qquad s \in dom(\overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\}) \qquad r \in dom((\overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\})(s))}{\cdot; \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \vdash_{\text{type}} s\sharp r} \qquad \cdot; \cdot; \mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \vdash_{\text{exp}} v : \forall\alpha.\mathsf{RGNHandle}\ \alpha \to \mathsf{RGN}\ \alpha\ \tau_a}{\cdot; \cdot; \mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \vdash_{\text{exp}} v\ [s\sharp r] : \mathsf{RGNHandle}\ s\sharp r \to \mathsf{RGN}\ s\sharp r\ \tau_a} \qquad \dfrac{\vdash_{\text{ctxt}} \cdot; \cdot; \mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \qquad s \in dom(\overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\}) \qquad r \in dom((\overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\})(s))}{\cdot; \cdot; \mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \vdash_{\text{exp}} \mathsf{handle}(s\sharp r) : \mathsf{RGNHandle}\ s\sharp r}}{\cdot; \cdot; \mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \vdash_{\text{exp}} v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) : \mathsf{RGN}\ s\sharp r\ \tau_a}$$

Applying the induction hypothesis to (1a) $\vdash_{\text{tower}} T, s \mapsto \cdot, r \mapsto \{\} : \mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\}$, (1b) $\cdot; \cdot; \mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \vdash_{\text{exp}} v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) : \mathsf{RGN}\ s\sharp r\ \tau_a$, and (1c) $T, s \mapsto \cdot, r \mapsto \{\}; v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) \hookrightarrow \kappa^{v'}$, we conclude $\cdot; \cdot; \mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \vdash_{\text{exp}} \kappa^{v'} : \mathsf{RGN}\ s\sharp r\ \tau_a$. Applying the induction hypothesis to (2a) $\vdash_{\text{tower}} T, s \mapsto \cdot, r \mapsto \{\} : \mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\}$, (2b) $\cdot; \cdot; \mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \vdash_{\text{exp}} \kappa^{v'} : \mathsf{RGN}\ s\sharp r\ \tau_a$, and (2c) $T, s \mapsto \cdot, r \mapsto \{\}; \kappa^v \hookrightarrow \cdot, r \mapsto R'''; v'''$, we conclude that there exists $\overline{\mathcal{S}}''' \supseteq_{=\supseteq} \cdot, r \mapsto \{\}$ and $\mathcal{S}''' \supseteq_{=\supseteq} \cdot, r \mapsto \{\}$ such that $\vdash_{\text{tower}} T, s \mapsto \cdot, r \mapsto R''' : \mathcal{T}, s \mapsto \mathcal{S}''' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}'''$ and $\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}''' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}''' \vdash_{\text{exp}} v''' : \tau_a$. Note that $\overline{\mathcal{S}}''' \supseteq_{=\supseteq} \cdot, r \mapsto \{\}$ implies $\overline{\mathcal{S}}''' \equiv \cdot, r \mapsto \overline{\mathcal{R}}'''$ and Note that $\mathcal{S}''' \supseteq_{=\supseteq} \cdot, r \mapsto \{\}$ implies $\mathcal{S}''' \equiv \cdot, r \mapsto \mathcal{R}'''$. Hence, we conclude that there exists $\overline{\mathcal{R}}''' \supseteq_{\supseteq} \{\}$ and $\mathcal{R}''' \supseteq_{\supseteq} \{\}$ such that $\vdash_{\text{tower}} T, s \mapsto \cdot, r \mapsto R''' : \mathcal{T}, s \mapsto \cdot, r \mapsto \mathcal{R}''' : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \overline{\mathcal{R}}'''$ and $\cdot; \cdot; \mathcal{T}, s \mapsto \cdot, r \mapsto \mathcal{R}''' : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \overline{\mathcal{R}}''' \vdash_{\text{exp}} v''' : \tau_a$. By Lemma 33, we conclude $\cdot; \cdot; \mathcal{T}, s \mapsto \cdot; \overline{\mathcal{T}}, s \mapsto \cdot \vdash_{\text{exp}} v'''[s\sharp \bullet / s\sharp r] : \tau_a[s\sharp \bullet / s\sharp r]$. By Lemma 34, we conclude $\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v'''[s\sharp \bullet / s\sharp r][\circ/s] : \tau_a[s\sharp \bullet / s\sharp r][\circ/s]$. By the derivation (1b), we conclude $\cdot; \overline{\mathcal{T}} \vdash_{\text{type}} \tau_a$. By Lemmas 36 and 37, we conclude $\tau_a[s\sharp \bullet / s\sharp r] = \tau_a$ and $\tau[s\sharp \bullet / s\sharp r][\circ/s] = \tau_a[\circ/s] = \tau_a$. Thus, $\cdot; \cdot; \mathcal{T} \vdash_{\text{exp}} v'''[s\sharp \bullet / s\sharp r][\circ/s] : \tau_a$, as required.

Case $\dfrac{}{T; \kappa^v \hookrightarrow \kappa^v}$: ...

Case $\dfrac{}{T; \langle l \rangle_{\sigma\sharp\rho} \hookrightarrow \langle l \rangle_{\sigma\sharp\rho}}$: ...

Case $\dfrac{}{T; \mathsf{handle}(\sigma\sharp\rho) \hookrightarrow \mathsf{handle}(\sigma\sharp\rho)}$: ...

Case $\dfrac{\tau_r \equiv s\sharp r}{T, s \mapsto S; \underline{\mathsf{returnRGN}}\ [\tau_r]\ [\tau_a]\ v \hookrightarrow_\kappa S; v}$: By inspection, the derivation of (2b) must end with

$$\dfrac{\cdot; \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{type}} s\sharp r \qquad \cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} v : \tau_a}{\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \underline{\mathsf{returnRGN}}\ [s\sharp r]\ [\tau_a]\ v : \mathsf{RGN}\ s\sharp r\ \tau_a}$$

We conclude $\overline{\mathcal{S}} \supseteq_{=\supseteq} \overline{\mathcal{S}}$, $\mathcal{S} \supseteq_{=\supseteq} \mathcal{S}$, and $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$ and $\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} v : \tau_a$, as required.

Case $\dfrac{\begin{array}{c}\tau_r \equiv s\sharp r \\ v_1 \equiv {\kappa^v}_1 \qquad T, s \mapsto S; {\kappa^v}_1 \hookrightarrow_\kappa S'; v_1' \\ T, s \mapsto S'; v_2\ v_1' \hookrightarrow v'' \qquad v'' \equiv \kappa^{v'''} \\ T, s \mapsto S'; \kappa^{v'''} \hookrightarrow_\kappa S'''; v'''\end{array}}{T, s \mapsto S; \underline{\mathsf{thenRGN}}\ [\tau_r]\ [\tau_a]\ [\tau_b]\ v_1\ v_2 \hookrightarrow_\kappa S'''; v'''}$: By inspection, the derivation of (2b) must end with

$$\dfrac{\begin{array}{c}\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v_1 : \mathsf{RGN}\ \tau_r\ \tau_a \\ \cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v_2 : \tau_a \to \mathsf{RGN}\ \tau_r\ \tau_b\end{array}}{\cdot; \cdot; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \underline{\mathsf{thenRGN}}\ [\tau_r]\ [\tau_a]\ [\tau_b]\ v_1\ v_2 : \mathsf{RGN}\ \tau_r\ \tau_b}$$

Applying the induction hypothesis to (2a) $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$, (2b) $\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} {\kappa^v}_1 : \mathsf{RGN}\ s\sharp r\ \tau_a$, and (2c) $T, s \mapsto S; {\kappa^v}_1 \hookrightarrow S'; v_1'$, we conclude that there exists $\overline{\mathcal{S}}' \supseteq_{=\supseteq} \overline{\mathcal{S}}$ and $\mathcal{S}' \supseteq_{=\supseteq} \mathcal{S}$ such that $\vdash_{\text{tower}} T, s \mapsto S' : \mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}'$ and $\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}' \vdash_{\text{exp}} v_1' : \tau_a$. By Lemma 29, we conclude $\vdash_{\text{ttype}} \mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}'$. By Lemma 32, we conclude $\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}' \vdash_{\text{exp}} v_2 : \tau_a \to \mathsf{RGN}\ s\sharp r\ \tau_b$. Note that

$$\dfrac{\boxed{\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}' \vdash_{\text{exp}} v_2 : \tau_a \to \mathsf{RGN}\ s\sharp r\ \tau_b} \qquad \boxed{\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}' \vdash_{\text{exp}} v_1' : \mathsf{RGN}\ s\sharp r\ \tau_a}}{\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}' \vdash_{\text{exp}} v_2\ v_1' : \mathsf{RGN}\ s\sharp r\ \tau_b}$$

Applying the induction hypothesis to (1a) $\vdash_{\text{tower}} T, s \mapsto S' : \mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}'$, (1b) $\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}' \vdash_{\text{exp}} v\ v' : \mathsf{RGN}\ s\sharp r\ \tau_b$, and (1c) $T, s \mapsto S'; v_2\ v_1' \hookrightarrow \kappa^{v''}$, we conclude $\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}' \vdash_{\text{exp}} \kappa^{v''} : \mathsf{RGN}\ s\sharp r\ \tau_b$. Applying the induction hypothesis to (2a) $\vdash_{\text{tower}} T, s \mapsto S' : \mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}'$, (2b) $\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}' \vdash_{\text{exp}} \kappa^{v''} : \mathsf{RGN}\ s\sharp r\ \tau_b$, and (2c) $T, s \mapsto S'; \kappa^{v''} \hookrightarrow S'''; v'''$, we conclude that there exists $\overline{\mathcal{S}}''' \supseteq_{=\supseteq} \overline{\mathcal{S}}'$ and $\mathcal{S}''' \supseteq_{=\supseteq} \mathcal{S}'$ such that $\vdash_{\text{tower}} T, s \mapsto S''' : \mathcal{T}, s \mapsto \mathcal{S}''' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}'''$ and $\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}''' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}''' \vdash_{\text{exp}} v''' : \tau_a$. By transitivity of $\supseteq_{=\supseteq}$, we conclude $\overline{\mathcal{S}}''' \supseteq_{=\supseteq} \overline{\mathcal{S}}$, $\mathcal{S}''' \supseteq_{=\supseteq} \mathcal{S}$, $\vdash_{\text{tower}} T, s \mapsto S''' : \mathcal{T}, s \mapsto \mathcal{S}''' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}'''$ and $\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}''' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}''' \vdash_{\text{exp}} v''' : \tau_a$, as required.

Case $\dfrac{\begin{array}{c}\tau_r \equiv s\sharp r_1 \\ r_1 \in dom(S) \qquad r_2 \notin dom(S) \\ T, s \mapsto S, r_2 \mapsto \{\}; v\ [s\sharp r_2]\ (\Lambda\beta.\lambda k : \mathsf{RGN}\ s\sharp r_1\ \beta.\underline{\mathsf{witnessRGN}}\ s\sharp r_1\ s\sharp r_2\ [\beta]\ k)\ \mathsf{handle}(s\sharp r_2) \hookrightarrow v' \qquad v' \equiv \kappa^{v'} \\ T, s \mapsto S, r_2 \mapsto \{\}; \kappa^{v'} \hookrightarrow_\kappa S''; v''' \qquad S'' \equiv S''', r_2 \mapsto R_2'''\end{array}}{T, s \mapsto S; \underline{\mathsf{letRGN}}\ [\tau_r]\ [\tau_a]\ v \hookrightarrow_\kappa S'''[s\sharp \bullet / s\sharp r_2]; v'''[s\sharp \bullet / s\sharp r_2]}$:
By inspection, the derivation of (2b) must end with

$$\dfrac{\begin{array}{c}\cdot; \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{type}} s\sharp r_1 \qquad \cdot; \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{type}} \tau_a \\ \cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} v : \forall\alpha.s\sharp r_1 \preceq \alpha \to \mathsf{RGNHandle}\ \alpha \to \mathsf{RGN}\ \alpha\ \tau_a\end{array}}{\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \underline{\mathsf{letRGN}}\ [s\sharp r_1]\ [\tau_a]\ v : \mathsf{RGN}\ \tau_r\ \tau_a}$$

By inspection of the derivation $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$, we conclude $\vdash_{\text{tdom}} \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$, $\vdash_{\text{ttype}} \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$, and $dom(S) = dom(\mathcal{S}) = dom(\overline{\mathcal{S}})$. Note that $\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \supseteq_{=\supseteq=} \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$ and $\mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} \supseteq_{=\supseteq=} \mathcal{T}, s \mapsto \mathcal{S}$ and

$$\dfrac{\boxed{\vdash_{\text{tdom}} \overline{\mathcal{T}}} \qquad \boxed{s \notin dom(\overline{\mathcal{T}})} \qquad \dfrac{\boxed{\vdash_{\text{sdom}} \overline{\mathcal{S}}} \qquad \boxed{r_2 \notin dom(\overline{\mathcal{S}})} \qquad \dfrac{}{\boxed{\vdash_{\text{rdom}} \{\}}}}{\vdash_{\text{sdom}} \overline{\mathcal{S}}, r_2 \mapsto \{\}}}{\vdash_{\text{tdom}} \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\}}$$

and $\vdash_{\text{ttype}} \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\}$ (requires appealing to Lemma 32) and $\vdash_{\text{tower}} T, s \mapsto S, r_2 \mapsto \{\} : \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\}$ (requires appealing to Lemma 32). By Lemma 32, we conclude $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} v : \forall \alpha. s\sharp r_1 \preceq \alpha \to \mathsf{RGNHandle}\ \alpha \to \mathsf{RGN}\ \alpha\ \tau_a$. Let $e_{\text{wit}} = \Lambda\beta.\lambda k : \mathsf{RGN}\ s\sharp r_1\ \beta.\underline{\mathsf{witnessRGN}}\ s\sharp r_1\ s\sharp r_2\ [\beta]\ k$. Note that

$$\dfrac{\dfrac{\dfrac{\vdash_{\text{ctxt}} \cdot, \beta; \cdot; k : \mathsf{RGN}\ s\sharp r_1\ \beta; \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\}}{\cdot, \beta; \cdot, k : \mathsf{RGN}\ s\sharp r_1\ \beta; \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} k : \mathsf{RGN}\ s\sharp r_1\ \beta} \quad \boxed{\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{cast}} s\sharp r_1 \rightsquigarrow s\sharp r_2}}{\cdot, \beta; \cdot, k : \mathsf{RGN}\ s\sharp r_1\ \beta; \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} \underline{\mathsf{witnessRGN}}\ s\sharp r_1\ s\sharp r_2\ [\beta]\ k : \mathsf{RGN}\ s\sharp r_2\ \beta}}{\cdot, \beta; \cdot; \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} \lambda k : \mathsf{RGN}\ s\sharp r_1\ \beta.\underline{\mathsf{witnessRGN}}\ s\sharp r_1\ s\sharp r_2\ [\beta]\ k : \mathsf{RGN}\ s\sharp r_1\ \beta \to \mathsf{RGN}\ s\sharp r_2\ \beta}}{\cdot;\cdot; \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} e_{\text{wit}} : s\sharp r_1 \preceq s\sharp r_2}$$

and

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\cdot;\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{type}} s\sharp r_2} \quad \boxed{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} v : \forall \alpha. s\sharp r_1 \preceq \alpha \to \mathsf{RGNHandle}\ \alpha \to \mathsf{RGN}\ \alpha\ \tau_a}}{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} v\ [s\sharp r_2] : s\sharp r_1 \preceq s\sharp r_2 \to \mathsf{RGNHandle}\ s\sharp r_2 \to \mathsf{RGN}\ s\sharp r_2\ \tau_a} \quad \boxed{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} e_{\text{wit}} : s\sharp r_1 \preceq s\sharp r_2}}{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} v\ [s\sharp r_2]\ e_{\text{wit}} : \mathsf{RGNHandle}\ s\sharp r_2 \to \mathsf{RGN}\ s\sharp r_2\ \tau_a} \quad \boxed{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} \mathsf{handle}(s\sharp r_2) : \mathsf{RGNHandle}\ s\sharp r_2}}{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} v\ [s\sharp r_2]\ e_{\text{wit}}\ \mathsf{handle}(s\sharp r_2) : \mathsf{RGN}\ s\sharp r_2\ \tau_a}$$

Applying the induction hypothesis to (1a) $\vdash_{\text{tower}} T, s \mapsto S, r_2 \mapsto \{\} : \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\}$, (1b) $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} v\ [s\sharp r_2]\ e_{\text{wit}}\ \mathsf{handle}(s\sharp r_2) : \mathsf{RGN}\ s\sharp r_2\ \tau_a$, and (1c) $T, s \mapsto S, r_2 \mapsto \{\}; v\ [s\sharp r_2]\ e_{\text{wit}}\ \mathsf{handle}(s\sharp r_2) \hookrightarrow \kappa^{v'}$, we conclude $;\cdot;\mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} \kappa^{v'} : \mathsf{RGN}\ s\sharp r_2\ \tau_a$. Applying the induction hypothesis to (2a) $\vdash_{\text{tower}} T, s \mapsto S, r_2 \mapsto \{\} : \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\}$, (2b) $;\cdot;\mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} \kappa^{v'} : \mathsf{RGN}\ s\sharp r_2\ \tau_a$, and (2c) $T, s \mapsto S, r_2 \mapsto \{\}; \kappa^{v'} \hookrightarrow_\kappa S''', r_2 \mapsto R_2'''; v'''$, we conclude that there exists $\overline{\mathcal{S}}^\dagger \supseteq_{=\supseteq} \overline{\mathcal{S}}, r_2 \mapsto \{\}$ and $\mathcal{S}^\dagger \supseteq_{=\supseteq} \mathcal{S}, r_2 \mapsto \{\}$ such that $\vdash_{\text{tower}} T, s \mapsto S''', r_2 \mapsto R_2''' : \mathcal{T}, s \mapsto \mathcal{S}^\dagger : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}^\dagger$ and $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}^\dagger : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}^\dagger \vdash_{\text{exp}} v''' : \tau_a$. Note that $\overline{\mathcal{S}}^\dagger \supseteq_{=\supseteq} \overline{\mathcal{S}}, r_2 \mapsto \{\}$ implies $\overline{\mathcal{S}}^\dagger \equiv \overline{\mathcal{S}}''', r_2 \mapsto \overline{\mathcal{R}}_2'''$. Note that $\mathcal{S}^\dagger \supseteq_{=\supseteq} \mathcal{S}, r_2 \mapsto \{\}$ implies $\mathcal{S}^\dagger \equiv \mathcal{S}''', r_2 \mapsto \mathcal{R}_2'''$. Hence, we conclude that there exists $\overline{\mathcal{S}}''' \supseteq_{=\supseteq} \overline{\mathcal{S}}$ and $\overline{\mathcal{R}}_2''' \supseteq_\supseteq \{\}$ and $\mathcal{S}''' \supseteq_{=\supseteq} \mathcal{S}$ and $\mathcal{R}_2''' \supseteq_\supseteq \{\}$ such that $\vdash_{\text{tower}} T, s \mapsto S''', r_2 \mapsto R_2''' : \mathcal{T}, s \mapsto \mathcal{S}''', r_2 \mapsto \overline{\mathcal{R}}_2''' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}''', r_2 \mapsto \overline{\mathcal{R}}_2'''$ and $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}''', r_2 \mapsto \mathcal{R}_2''' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}''', r_2 \mapsto \overline{\mathcal{R}}_2''' \vdash_{\text{exp}} v''' : \tau_a$. By Lemma 33, we conclude $\vdash_{\text{tower}} T, s \mapsto S'''[s\sharp \bullet / s\sharp r_2] : \mathcal{T}, s \mapsto \mathcal{S}'''[s\sharp \bullet / s\sharp r_2]$. By Lemma 33, we conclude $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}'''[s\sharp \bullet / s\sharp r_2] \vdash_{\text{exp}} v'''[s\sharp \bullet / s\sharp r_2] : \tau_a[s\sharp \bullet / s\sharp r_2]$. By the derivation (2b), we conclude $\cdot;\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{type}} \tau_a$. By Lemma 36, we conclude $\tau_a[s\sharp \bullet / s\sharp r_2] = \tau_a$. Note that $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$ implies $\forall r \in dom(\mathcal{S}). \forall l \in dom(\mathcal{S}(r)). \cdot; \mathcal{T}, s \mapsto \mathcal{S} \vdash_{\text{type}} \mathcal{S}(r, l)$. Note that $\mathcal{S}''' \supseteq_{=\supseteq} \mathcal{S}$ implies $\forall r \in dom(\mathcal{S}). \forall l \in dom(\mathcal{S}(r)). \mathcal{S}'''(r, l) = \mathcal{S}(r, l)$. Hence, $\forall r \in dom(\mathcal{S}). \forall l \in dom(\mathcal{S}(r)). \cdot; \mathcal{T}, s \mapsto \mathcal{S} \vdash_{\text{type}} \mathcal{S}'''(r, l)$. By Lemma 36 (using $r_2 \notin dom(\mathcal{S})$), we conclude $\forall r \in dom(\mathcal{S}). \forall l \in dom(\mathcal{S}(r)). \mathcal{S}'''(r, l)[s\sharp \bullet / s\sharp r_2] \equiv \mathcal{S}'''(r, l)$. Furthermore, $\forall r \in dom(\mathcal{S}). \forall l \in dom(\mathcal{S}(r)). \mathcal{S}'''[s\sharp \bullet / s\sharp r_2](r, l) \equiv \mathcal{S}'''(r, l)$. Hence, $\forall r \in dom(\mathcal{S}). \forall l \in dom(\mathcal{S}(r)). \mathcal{S}'''[s\sharp \bullet / s\sharp r_2](r, l) = \mathcal{S}(r, l)$. Hence, $\mathcal{S}'''[s\sharp \bullet / s\sharp r_2] \supseteq_{=\supseteq} \mathcal{S}$. Hence, $\overline{\mathcal{S}}''' \supseteq_{=\supseteq} \overline{\mathcal{S}}$, $\mathcal{S}'''[s\sharp \bullet / s\sharp r_2] \supseteq_{=\supseteq} \mathcal{S}$, $\vdash_{\text{tower}} T, s \mapsto S'''[s\sharp \bullet / s\sharp r_2] : \mathcal{T}, s \mapsto \mathcal{S}'''[s\sharp \bullet / s\sharp r_2] : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}'''$, and $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}'''[s\sharp \bullet / s\sharp r_2] : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}''' \vdash_{\text{exp}} v'''[s\sharp \bullet / s\sharp r_2] : \tau_a$, as required.

Case $\dfrac{\begin{array}{c}\sigma\sharp\rho_1 \equiv s\sharp r_1 \qquad \sigma\sharp\rho_2 \equiv s\sharp r_2 \qquad v \equiv \kappa^v \\ S \equiv S_1, r_1 \mapsto R_1, S_2, r_2 \mapsto R_2, S_3 \\ T, s \mapsto S; \kappa^v \hookrightarrow_\kappa S'; v'\end{array}}{T, s \mapsto S; \underline{\mathsf{witnessRGN}}\ \sigma\sharp\rho_1\ \sigma\sharp\rho_2\ [\tau_a]\ v \hookrightarrow_\kappa S'; v'}$: By inspection, the derivation of (2b) must end with

$$\dfrac{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} \vdash_{\text{exp}} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}\kappa^v : \mathsf{RGN}\ s\sharp r_1\ \tau_a \qquad \dfrac{\overline{\mathcal{S}} = \overline{\mathcal{S}}_1, r_1 \mapsto \overline{\mathcal{R}}_1, \overline{\mathcal{S}}_2, r_2 \mapsto \overline{\mathcal{R}}_2, \overline{\mathcal{S}}_3}{\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{cast}} s\sharp r_1 \rightsquigarrow s\sharp r_2}}{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \underline{\mathsf{witnessRGN}}\ s\sharp r_1\ s\sharp r_2\ [\tau_a]\ \kappa^v : \mathsf{RGN}\ s\sharp r_2\ \tau_a}$$

Applying the induction hypothesis to (2a) $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$, (2b) $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \kappa^v : \mathsf{RGN}\ s\sharp r_1\ \tau_a$, and (2c) $T, s \mapsto S; \kappa^v \hookrightarrow S'; v'$, we conclude that there exists $\overline{\mathcal{S}}' \supseteq_{=\supseteq} \overline{\mathcal{S}}$ and $\mathcal{S}' \supseteq_{=\supseteq} \mathcal{S}$ such that $\vdash_{\text{tower}} T, s \mapsto S' : \mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}'$ and $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}' \vdash_{\text{exp}} v' : \tau_a$, as required.

Case $\dfrac{\begin{array}{c}\tau_r \equiv s\sharp r \qquad v_1 \equiv \mathsf{handle}(s\sharp r) \\ r \in dom(S) \qquad l \notin dom(S(r))\end{array}}{T, s \mapsto S; \underline{\mathsf{newRGNVar}}\ [\tau_r]\ [\tau_a]\ v_1\ v_2 \hookrightarrow_\kappa S\{(r,l) \mapsto v_2\}; \langle l \rangle_{s\sharp r}}$: By inspection, the derivation of (2b) must end with

$$\dfrac{\dfrac{s \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}) \qquad r \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})(s))}{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \mathsf{handle}(s\sharp r) : \mathsf{RGNHandle}\ s\sharp r} \qquad \cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} w : \tau_a}{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \underline{\mathsf{newRGNVar}}\ [s\sharp r]\ [\tau_a]\ \mathsf{handle}(s\sharp r)\ w : \mathsf{RGN}\ s\sharp r\ (\mathsf{RGNVar}\ s\sharp r\ \tau_a)}$$

Note that $\overline{\mathcal{S}}\{(r,l) \mapsto\} \supseteq_{=\supseteq} \mathcal{S}$ and $\mathcal{S}\{(r,l) \mapsto \tau_a\} \supseteq_{=\supseteq} \mathcal{S}$. Note that $\vdash_{\text{tower}} T, s \mapsto S\{(r,l) \mapsto w\} : \mathcal{T}, s \mapsto \mathcal{S}\{(r,l) \mapsto \tau_a\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}\{(r,l) \mapsto\}$ (required appealing to Lemma 32). Note that

$$\dfrac{\dfrac{\boxed{\vdash_{\text{ttype}} \mathcal{T}, s \mapsto \mathcal{S}\{(r,l) \mapsto \tau_a\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}\{(r,l) \mapsto\}} \qquad \boxed{\vdash_{\text{tctxt}} \cdot} \qquad \dfrac{\boxed{\vdash_{\text{tdom}} \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}\{(r,l) \mapsto\}} \qquad \boxed{\vdash_{\text{tctxt}} \cdot}}{\cdot; \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}\{(r,l) \mapsto\} \vdash_{\text{vctxt}} \cdot}}{\vdash_{\text{ctxt}} \cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}\{(r,l) \mapsto \tau_a\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}\{(r,l) \mapsto\}} \qquad \begin{array}{l}\boxed{s \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}\{(r,l) \mapsto\})} \\ \boxed{l \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}\{(r,l) \mapsto\}))}\end{array} \qquad \begin{array}{l}\boxed{r \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}\{(r,l) \mapsto\})(s))} \\ \boxed{\tau_a = (\mathcal{T}, s \mapsto \mathcal{S}\{(r,l) \mapsto \tau_a\})(s,r,l)}\end{array}}{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}\{(r,l) \mapsto \tau_a\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}\{(r,l) \mapsto\} \vdash_{\text{exp}} \langle l \rangle_{s\sharp r} : \mathsf{RGNVar}\ s\sharp r\ \tau_a}$$

Hence, $\overline{\mathcal{S}}\{(r,l) \mapsto\} \supseteq_{=\supseteq} \mathcal{S}$, $\mathcal{S}\{(r,l) \mapsto \tau_a\} \supseteq_{=\supseteq} \mathcal{S}$, $\vdash_{\text{tower}} T, s \mapsto S\{(r,l) \mapsto w\} : \mathcal{T}, s \mapsto \mathcal{S}\{(r,l) \mapsto \tau_a\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}\{(r,l) \mapsto\}$, and $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}\{(r,l) \mapsto \tau_a\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}\{(r,l) \mapsto\} \vdash_{\text{exp}} \langle l \rangle_{s\sharp r} : \mathsf{RGNVar}\ s\sharp r\ \tau_a$, as required.

Case $\dfrac{\begin{array}{ccc}\tau_r \equiv s\sharp r & & v \equiv \langle l \rangle_{s\sharp r} \\ r \in dom(S) & l \in dom(S(r)) & v' = S(r,l)\end{array}}{T, s \mapsto S; \underline{\mathsf{readRGNVar}}\ [\tau_r]\ [\tau_a]\ v \hookrightarrow_\kappa S; v'}$: By inspection, the derivation of (2b) must end with

$$\dfrac{\begin{array}{cc}s \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}) & r \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})(s)) \\ l \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})(s,r)) & \tau_a = (\mathcal{T}, s \mapsto \mathcal{S})(s,r,l)\end{array}}{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \langle l \rangle_{s\sharp r} : \mathsf{RGNVar}\ s\sharp r\ \tau_a}{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} \vdash_{\text{exp}} \underline{\mathsf{readRGNVar}}\ [s\sharp r]\ [\tau_a]\ \langle l \rangle_{s\sharp r} : \mathsf{RGN}\ s\sharp r\ \tau_a}$$

By $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$, $r \in dom(S)$, and $l \in dom(S(r))$, we conclude $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} \vdash_{\text{exp}} S(r,l) : \mathcal{S}(r,l)$. Hence, $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} \vdash_{\text{exp}} w : \tau_a$. We conclude $\overline{\mathcal{S}} \supseteq_{=\supseteq} \overline{\mathcal{S}}$, $\mathcal{S} \supseteq_{=\supseteq} \mathcal{S}$, $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$, and $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} w : \tau_a$, as required.

□

3.5.3 Progress

Definition 1

(1) A pending judgement $T; e \hookrightarrow ?$ is well typed iff there exists $\overline{\mathcal{T}}$, $\mathcal{T}$, and τ such that $\vdash_{\text{tower}} T : \mathcal{T} : \overline{\mathcal{T}}$ and $\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e : \tau$.

(2) A pending judgement $T, s \mapsto S; \kappa^v \hookrightarrow ?$ is well typed iff there exists $\overline{\mathcal{T}}$, $\mathcal{T}$, $\overline{\mathcal{S}}$, $\mathcal{S}$, $r \in dom(\overline{\mathcal{S}})$, and τ_a such that $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$ and $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \kappa^v : \mathsf{RGN}\ s\sharp r\ \tau_a$.

(3) A partial derivation $\mathfrak{D}$ is well typed iff every pending judgement in it is well typed.

Theorem 3 (Progress)

If $\mathfrak{D}$ is a well-typed partial derivation with pending judgements, then there exists $\mathfrak{D}'$ such that $\mathfrak{D} \longrightarrow \mathfrak{D}'$ and $\mathfrak{D}'$ is well typed.

Proof. Let $\mathfrak{N}$ be the uppermost node of $\mathfrak{D}$ that is labeled with a pending judgement, either $T; e \hookrightarrow ?$ or $T, s \mapsto S; \kappa^v \hookrightarrow_\kappa ?$. Any transition on $\mathfrak{D}$ must occur at this node. We consider all posible forms of pending judgments.

Case $T; i \hookrightarrow ?$: ...

Case $T; e_1 \oplus e_2 \hookrightarrow ?$: ...

Case $T; e_1 \otimes e_2 \hookrightarrow ?$: ...

Case $T; \mathsf{tt} \hookrightarrow ?$: ...

Case $T; \mathsf{ff} \hookrightarrow ?$: ...

Case $T; \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f \hookrightarrow ?$: ...

Case $T; \lambda x : \tau . e \hookrightarrow ?$: ...

Case $T; e_1\ e_2 \hookrightarrow ?$: Because $\mathfrak{D}$ is well typed, $T; e_1\ e_2 \hookrightarrow ?$ is well typed. Hence, there exists $\overline{\mathcal{T}}$, $\mathcal{T}$, and τ_2 such that $\vdash_{\text{tower}} T : \mathcal{T} : \overline{\mathcal{T}}$ and $\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_1\ e_2 : \tau_2$. By inspection, the latter derivation must end with

$$\frac{\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_1 : \tau_1 \to \tau_2 \qquad \cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_2 : \tau_1}{\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_1\ e_2 : \tau_2}$$

Consider the children of $T; e_1\ e_2 \hookrightarrow ?$:

Case $()$: Add $[T; e_1 \hookrightarrow ?]()$, which is well typed by $\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_1 : \tau_1 \to \tau_2$.

Case $([T; e_1 \hookrightarrow v_1])$: Applying Preservation to (1a) $\vdash_{\text{tower}} T : \mathcal{T} : \overline{\mathcal{T}}$, (1b) $\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_1 : \tau_1 \to \tau_2$, and (1c) $T; e_1 \hookrightarrow v_1$, we conclude $\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v_1 : \tau_1 \to \tau_2$. By Lemma 38, we conclude $v_1 \equiv \lambda x : \tau_1 . e'$. Add $[v_1 \equiv \lambda x : \tau_1 . e'_1]$.

Case $([T; e_1 \hookrightarrow v_1], [v_1 \equiv \lambda x : \tau_1 . e'_1])$: Add $[T; e_2 \hookrightarrow ?]()$, which is well typed by $\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_2 : \tau_1$.

Case $([T; e_1 \hookrightarrow v_1], [v_1 \equiv \lambda x : \tau_1 . e'], [T; e_2 \hookrightarrow v_2])$: Applying Preservation to (1a) $\vdash_{\text{tower}} T : \mathcal{T} : \overline{\mathcal{T}}$, (1b) $\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_2 : \tau_1$, and (1c) $T; e_2 \hookrightarrow v_2$, we conclude $\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v_2 : \tau_1$. Recall, $\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \lambda x : \tau_1 . e'_1 : \tau_1 \to \tau_2$. By inspection, this derivation must end with

$$\frac{\cdot;\cdot, x : \tau_1;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e'_1 : \tau_2}{\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \lambda x : \tau_1 . e'_1 : \tau_1 \to \tau_2}$$

By Lemma 31, we conclude $\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e'_1[v_2/x] : \tau_2$. Add $[T; e'_1[v_2/x] \hookrightarrow ?]()$, which is well typed by $\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e'_1[v_2/x] : \tau_2$.

Case $([T; e_1 \hookrightarrow v_1], [v_1 \equiv \lambda x : \tau_1 . e'_1], [T; e_2 \hookrightarrow v_2], [T; e'_1[v_2/x] \hookrightarrow v_3])$: Replace $\mathfrak{N}$ with $[T; e_1\ e_2 \hookrightarrow v_3]$.

Case $T; (e_1, \ldots, e_n) \hookrightarrow ?$: ...

Case $T; \mathsf{sel}_i\ e \hookrightarrow ?$: ...

Case $T; \Lambda\alpha.e \hookrightarrow ?$: ...

Case $T; e\ [\tau] \hookrightarrow ?$: ...

Case $T; \mathsf{let}\ x = e_1\ \mathsf{in}\ e_2 \hookrightarrow ?$: ...

Case $T; \mathsf{runRGN}\ [\tau_a]\ v \hookrightarrow ?$: Because $\mathfrak{D}$ is well typed, $T; \mathsf{runRGN}\ [\tau_a]\ v \hookrightarrow ?$ is well typed. Hence, there exists $\overline{\mathcal{T}}$, $\mathcal{T}$, and $\tau \equiv \tau_a$ such that $\vdash_{\text{tower}} T : \mathcal{T} : \overline{\mathcal{T}}$ and $\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \mathsf{runRGN}\ [\tau_a]\ v : \tau_a$. By inspection, the latter derivation must end with

$$\frac{\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v : \forall\alpha.\mathsf{RGNHandle}\ \alpha \to \mathsf{RGN}\ \alpha\ \tau_a}{\cdot;\cdot;\mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \mathsf{runRGN}\ [\tau_a]\ v : \tau_a}$$

Consider the children of $T; \mathsf{runRGN}\ [\tau_a]\ v \hookrightarrow ?$:

Case $()$: Note that there exists $s \notin dom(T)$. Add $[s \notin dom(T)]$.

Case $([s \notin dom(T)])$: Note that $\overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \supseteq_{\supseteq==} \overline{\mathcal{T}}$ and $\mathcal{T}, s \mapsto \cdot, r \mapsto \{\} \supseteq_{\supseteq==} \mathcal{T}$ and

$$\frac{\vdash_{\text{tdom}} \overline{\mathcal{T}} \qquad s \notin dom(\overline{\mathcal{T}}) \qquad \dfrac{\vdash_{\text{sdom}} \cdot \qquad r \notin dom(\cdot) \qquad \vdash_{\text{rdom}} \{\}}{\vdash_{\text{sdom}} \cdot, r \mapsto \{\}}}{\vdash_{\text{tdom}} \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\}}$$

and $\vdash_{\text{ttype}} \mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\}$ (requires appealing to Lemma 32) and $\vdash_{\text{tower}} T, s \mapsto \cdot, r \mapsto \{\} : \mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\}$ (requires appaling to Lemma 32). By Lemma 32, we conclude $\cdot;\cdot;\mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \vdash_{\text{exp}} v : \forall\alpha.\mathsf{RGNHandle} \to \mathsf{RGN}\ \alpha\ \tau_a$. Note that

$$\frac{\dfrac{\vdash_{\text{tdom}} \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \qquad \vdash_{\text{tctxt}} \cdot \qquad s \in dom(\overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\}) \qquad r \in dom((\overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\})(s))}{\cdot; \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \vdash_{\text{type}} s\sharp r} \qquad \vdash_{\text{exp}} v : \forall\alpha.\mathsf{RGNHandle}\ \alpha \to \mathsf{RGN}\ \alpha\ \tau_a}{\cdot;\cdot;\mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \vdash_{\text{exp}} v\ [s\sharp r] : \mathsf{RGNHandle}\ s\sharp r \to \mathsf{RGN}\ s\sharp r\ \tau_a} \qquad \dfrac{\vdash_{\text{ctxt}} \cdot;\cdot;\mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \qquad s \in dom(\overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\}) \qquad r \in dom((\overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\})(s))}{\cdot;\cdot;\mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \vdash_{\text{exp}} \mathsf{handle}(s\sharp r) : \mathsf{RGNHandle}\ s\sharp r}}{\cdot;\cdot;\mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \vdash_{\text{exp}} v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) : \mathsf{RGN}\ s\sharp r\ \tau_a}$$

Add $[T, s \mapsto \cdot, r \mapsto \{\}; v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) \hookrightarrow ?]()$, which is well typed by $\cdot;\cdot;\mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \vdash_{\text{exp}} v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) : \mathsf{RGN}\ s\sharp r\ \tau_a$.

Case $([s \notin dom(T)], [T, s \mapsto \cdot, r \mapsto \{\}; v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) \hookrightarrow v'])$: Applying Preservation to (1a) $\vdash_{\text{tower}} T, s \mapsto \cdot, r \mapsto \{\} \vdash_{\text{exp}} \mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\}$, (1b) $\cdot;\cdot;\mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \vdash_{\text{exp}} v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) : \mathsf{RGN}\ s\sharp r\ \tau_a$, and (1c) $T, s \mapsto \cdot, r \mapsto \{\}; v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) \hookrightarrow v'$, we conclude $\cdot;\cdot;\mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \vdash_{\text{exp}} v' : \mathsf{RGN}\ s\sharp r\ \tau_a$. By Lemma 38, we conclude $v' \equiv \kappa^{v'}$. Add $[v' \equiv \kappa^{v'}]$.

Case $([s \notin dom(T)], [T, s \mapsto \cdot, r \mapsto \{\}; v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) \hookrightarrow v'], [v' \equiv \kappa^{v'}])$: Add $[T, s \mapsto \cdot, r \mapsto \{\}; \kappa^{v'} \hookrightarrow_\kappa ?]()$, which is well typed by $\cdot;\cdot;\mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \vdash_{\text{exp}} \kappa^{v'} : \mathsf{RGN}\ s\sharp r\ \tau_a$.

Case $([s \notin dom(T)], [T, s \mapsto \cdot, r \mapsto \{\}; v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) \hookrightarrow v'], [v' \equiv \kappa^{v}], [T, s \mapsto \cdot, r \mapsto \{\}; \kappa^{v'} \hookrightarrow_\kappa S''; v''']):$ Applying Preservation to (2a) $\vdash_{\text{tower}} T, s \mapsto \cdot, r \mapsto \{\} : \mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\}$, (2b) $\cdot;\cdot;\mathcal{T}, s \mapsto \cdot, r \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \{\} \vdash_{\text{exp}} \kappa^{v'} : \mathsf{RGN}\ s\sharp r\ \tau_a$, and (2c) $T, s \mapsto \cdot, r \mapsto \{\}; \kappa^{v'} \hookrightarrow_\kappa S''; v'''$, we conclude that there exists $\overline{\mathcal{S}}'' \supseteq_{=\supseteq} \cdot, r \mapsto \{\}$ and $\mathcal{S}'' \supseteq_{=\supseteq} \cdot, r \mapsto \{\}$ such that $\vdash_{\text{tower}} T, s \mapsto S'' : \mathcal{T}, s \mapsto \mathcal{S}'' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}''$ and $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}'' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}'' \vdash_{\text{exp}} v''' : \tau_a$. Note that $\overline{\mathcal{S}}'' \supseteq_{=\supseteq} \cdot, r \mapsto \{\}$ implies $\overline{\mathcal{S}}'' \equiv \cdot, r \mapsto \overline{\mathcal{R}}''$. Note that $\mathcal{S}'' \supseteq_{=\supseteq} \cdot, r \mapsto \{\}$ implies $\mathcal{S}'' \equiv \cdot, r \mapsto \mathcal{R}''$. Note that $S'' \equiv \cdot, r \mapsto R''$, because $\vdash_{\text{tower}} T, s \mapsto S'' : \mathcal{T}, s \mapsto \cdot, r \mapsto \mathcal{R}'' : \overline{\mathcal{T}}, s \mapsto \cdot, r \mapsto \overline{\mathcal{R}}''$. Add $[S'' \equiv \cdot, r \mapsto R'']$.

Case $([s \notin dom(T)], [T, s \mapsto \cdot, r \mapsto \{\}; v\ [s\sharp r]\ \mathsf{handle}(s\sharp r) \hookrightarrow v'], [v' \equiv \kappa^{v\prime}], [T, s \mapsto \cdot, r \mapsto \{\}; \kappa^{v\prime} \hookrightarrow_\kappa S''; v'''], [S'' \equiv \cdot, r \mapsto R'''])$: Replace $\mathfrak{N}$ with $[T; \mathsf{runRGN}\ [\tau_a]\ v \hookrightarrow v'''[s\sharp \bullet / s\sharp r][\circ/s]]$.

Case $T; \kappa^v \hookrightarrow ?$: ...

Case $T; \langle l \rangle_{\sigma\sharp\rho} \hookrightarrow ?$: ...

Case $T; \mathsf{handle}(\sigma\sharp\rho) \hookrightarrow ?$: ...

Case $T; \underline{\mathsf{returnRGN}}\ [\tau_r]\ [\tau_a]\ v \hookrightarrow_\kappa ?$: Because $\mathfrak{D}$ is well typed, $T, s \mapsto S; \underline{\mathsf{returnRGN}}\ [\tau_r]\ [\tau_a]\ v \hookrightarrow_\kappa ?$ is well typed. Hence, there exists $\overline{\mathcal{T}}$, $\mathcal{T}$, $\overline{\mathcal{S}}$, $\mathcal{S}$, $r \in dom(\overline{\mathcal{S}})$, and τ_a such that $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$ and $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \underline{\mathsf{returnRGN}}\ [\tau_r]\ [\tau_a]\ v : \mathsf{RGN}\ s\sharp r\ \tau_a$. By inspection, the latter derivation must end with

$$\frac{\cdot;\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{type}} s\sharp r \qquad \cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} v : \tau_a}{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \underline{\mathsf{returnRGN}}\ [s\sharp r]\ [\tau_a]\ v : \mathsf{RGN}\ s\sharp r\ \tau_a}$$

and $\tau_r \equiv s\sharp r$. Consider the children of $T, s \mapsto S; \underline{\mathsf{returnRGN}}\ [\tau_r]\ [\tau_a]\ v \hookrightarrow_\kappa ?$:

Case $()$: Add $[\tau_r \equiv s\sharp r]$.

Case $([\tau_r \equiv s\sharp r])$: Replace $\mathfrak{N}$ with $[T, s \mapsto S; \underline{\mathsf{returnRGN}}\ [s\sharp r]\ [\tau_a]\ v \hookrightarrow_\kappa S; v]$.

Case $T; \underline{\mathsf{thenRGN}}\ [\tau_r]\ [\tau_a]\ [\tau_b]\ v_1\ v_2 \hookrightarrow_\kappa ?$: Because $\mathfrak{D}$ is well typed, $T, s \mapsto S; \underline{\mathsf{thenRGN}}\ [\tau_r]\ [\tau_a]\ [\tau_b]\ v_1\ v_2 \hookrightarrow_\kappa ?$ is well typed. Hence, there exists $\overline{\mathcal{T}}$, $\mathcal{T}$, $\overline{\mathcal{S}}$, $\mathcal{S}$, $r \in dom(\overline{\mathcal{S}})$, and τ_a such that $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$ and $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \underline{\mathsf{thenRGN}}\ [\tau_r]\ [\tau_a]\ [\tau_b]\ v_1\ v_2 : \mathsf{RGN}\ s\sharp r\ \tau_a$. By inspection, the latter derivation must end with

$$\frac{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} v_1 : \mathsf{RGN}\ s\sharp r\ \tau_a \qquad \cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} v_2 : \tau_a \to \mathsf{RGN}\ s\sharp r\ \tau_b}{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \underline{\mathsf{thenRGN}}\ [s\sharp r]\ [\tau_a]\ [\tau_b]\ v_1\ v_2 : \mathsf{RGN}\ s\sharp r\ \tau_a}$$

and $\tau_r \equiv s\sharp r$. Consider the children of $T, s \mapsto S; \underline{\mathsf{thenRGN}}\ [\tau_r]\ [\tau_a]\ [\tau_b]\ v_1\ v_2 \hookrightarrow_\kappa ?$:

Case $()$: Add $[\tau_r \equiv s\sharp r]$.

Case $([\tau_r \equiv s\sharp r])$: By Lemma 38, we conclude $v_1 \equiv \kappa^v{}_1$. Add $[v_1 \equiv \kappa^v{}_1]$.

Case $([\tau_r \equiv s\sharp r], [v_1 \equiv \kappa^v{}_1])$: Add $[T, s \mapsto S; \kappa^v{}_1 \hookrightarrow_\kappa ?]()$, which is well typed by $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} v_1 : \mathsf{RGN}\ s\sharp r\ \tau_a$.

Case $([\tau_r \equiv s\sharp r], [v_1 \equiv \kappa^v{}_1], [T, s \mapsto S; \kappa^v{}_1 \hookrightarrow_\kappa S'; v_1'])$: Applying Preservation to (2a) $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$, (2b) $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \kappa^v{}_1 : \mathsf{RGN}\ s\sharp r\ \tau_a$, and (2c) $T, s \mapsto S; \kappa^v{}_1 \hookrightarrow_\kappa S'; v_1'$, we conclude that there exists $\overline{\mathcal{S}'} \supseteq_= \overline{\mathcal{S}}$ and $\mathcal{S}' \supseteq_= \mathcal{S}$ such that $\vdash_{\text{tower}} T, s \mapsto S' : \mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}'$ and $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}' \vdash_{\text{exp}} v_1' : \tau_a$. By Lemma 32, we conclude $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}' \vdash_{\text{exp}} v_2 : \tau_a \to \mathsf{RGN}\ s\sharp r\ \tau_b$. Note

$$\frac{\boxed{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}' \vdash_{\text{exp}} v_2 : \tau_a \to \mathsf{RGN}\ s\sharp r\ \tau_b} \qquad \boxed{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}' \vdash_{\text{exp}} v_1' : \tau_a}}{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}' \vdash_{\text{exp}} v_2\ v_1' : \mathsf{RGN}\ s\sharp r\ \tau_b}$$

Add $[T, s \mapsto S'; v_2\ v_1' \hookrightarrow ?]()$, which is well typed by $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}' \vdash_{\text{exp}} v_2\ v_1' : \mathsf{RGN}\ s\sharp r\ \tau_b$.

Case $([\tau_r \equiv s\sharp r], [v_1 \equiv \kappa^v{}_1], [T, s \mapsto S; \kappa^v{}_1 \hookrightarrow_\kappa S'; v'], [T, s \mapsto S'; v_2\ v_1' \hookrightarrow v''])$: Applying Preservation to (1a) $\vdash_{\text{tower}} T, s \mapsto S' : \mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}'$, (1b) $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}' \vdash_{\text{exp}} v_2\ v_1' : \mathsf{RGN}\ s\sharp r\ \tau_b$, and (1c) $T, s \mapsto S'; v_2\ v_1' \hookrightarrow v''$, we conclude $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}' \vdash_{\text{exp}} v'' : \mathsf{RGN}\ s\sharp r\ \tau_b$. By Lemma 38, we conclude $v'' \equiv \kappa^{v\prime\prime}$. Add $[v'' \equiv \kappa^{v\prime\prime}]$.

Case $([\tau_r \equiv s\sharp r], [v_1 \equiv \kappa^v{}_1], [T, s \mapsto S; \kappa^v{}_1 \hookrightarrow_\kappa S'; v'], [T, s \mapsto S'; v_2\ v_1' \hookrightarrow_\kappa v''], [v'' \equiv \kappa^{v\prime\prime}])$: Add $[T, s \mapsto S'; \kappa^{v\prime\prime} \hookrightarrow_\kappa ?]()$, which is well typed by $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}' \vdash_{\text{exp}} \kappa^{v\prime\prime} : \mathsf{RGN}\ s\sharp r\ \tau_b$.

Case $([\tau_r \equiv s\sharp r], [v_1 \equiv \kappa^v{}_1], [T, s \mapsto S; \kappa^v{}_1 \hookrightarrow_\kappa S'; v'], [T, s \mapsto S'; v_2\ v_1' \hookrightarrow_\kappa v''], [v'' \equiv \kappa^{v\prime\prime}], [T, s \mapsto S'; \kappa^{v\prime\prime} \hookrightarrow_\kappa S'''; v'''])$: Replace $\mathfrak{N}$ with $[T, s \mapsto S; \underline{\mathsf{thenRGN}}\ [s\sharp r]\ [\tau_a]\ [\tau_b]\ \kappa^v{}_1\ v_2 \hookrightarrow_\kappa S'''; v''']$.

Case $T, s \mapsto S; \underline{\mathsf{letRGN}}\ [\tau_r]\ [\tau_a]\ v \hookrightarrow_\kappa ?$: Because $\mathfrak{D}$ is well typed, $T, s \mapsto S; \underline{\mathsf{letRGN}}\ [\tau_r]\ [\tau_a]\ v \hookrightarrow_\kappa ?$ is well typed. Hence, there exists $\overline{\mathcal{T}}$, $\mathcal{T}$, $\overline{\mathcal{S}}$, $\mathcal{S}$, $r_1 \in dom(\mathcal{S})$, and τ_a such that $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$ and $\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S} \vdash_{\text{exp}} \underline{\mathsf{letRGN}}\ [\tau_r]\ [\tau_a]\ v : \mathsf{RGN}\ s\sharp r_1\ \tau_a$. By inspection, the latter derivation must end with

$$\frac{\cdot; \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{type}} \tau_a \qquad \dfrac{\cdot; \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{type}} s\sharp r_1 \qquad \cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} v : \forall \alpha.\mathsf{RGN}\ s\sharp r_1 \preceq \alpha \to \mathsf{RGNHandle}\ \alpha \to \mathsf{RGN}\ \alpha\ \tau_a}{\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \underline{\mathsf{letRGN}}\ [s\sharp r_1]\ [\tau_a]\ v : \mathsf{RGN}\ s\sharp r_1\ \tau_a}}{\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \underline{\mathsf{letRGN}}\ [s\sharp r_1]\ [\tau_a]\ v : \mathsf{RGN}\ s\sharp r_1\ \tau_a}$$

and $\tau_r \equiv s\sharp r_1$. Consider the children of $T, s \mapsto S; \underline{\mathsf{letRGN}}\ [\tau_r]\ [\tau_a]\ v \hookrightarrow_\kappa ?$:

Case $()$: Add $[\tau_r \equiv s\sharp r_1]$.

Case $([\tau_r \equiv s\sharp r_1])$: Note that $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$ and $r_1 \in dom(\overline{\mathcal{S}})$ implies $r_1 \in dom(S)$. Add $[r_1 \in dom(S)]$.

Case $([\tau_r \equiv s\sharp r_1], [r_1 \in dom(S)])$: Note that there exists $r_2 \notin dom(S)$. Add $[r_2 \notin dom(S)]$.

Case $([\tau_r \equiv s\sharp r_1], [r_1 \in dom(S)], [r_2 \notin dom(S)])$: Note that $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$ and $r_2 \notin dom(S)$ implies $r_2 \notin dom(\overline{\mathcal{S}})$. Note that $\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \supseteq_{=\supseteq=} \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$ and $\mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} \supseteq_{=\supseteq=} \mathcal{T}, s \mapsto \mathcal{S}$ and

$$\frac{\vdash_{\text{tdom}} \overline{\mathcal{T}} \qquad s \notin dom(\overline{\mathcal{T}}) \qquad \dfrac{\vdash_{\text{sdom}} \overline{\mathcal{S}} \qquad r_2 \notin dom(\overline{\mathcal{S}}) \qquad \dfrac{}{\vdash_{\text{rdom}} \{\}}}{\vdash_{\text{sdom}} \overline{\mathcal{S}}, r_2 \mapsto \{\}}}{\vdash_{\text{tdom}} \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\}}$$

and $\vdash_{\text{ttype}} \mathcal{T}, \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\}$ (requires appealing to Lemma 32) and $\vdash_{\text{tower}} T, s \mapsto S, r_2 \mapsto \{\} : \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\}$ (requires appealing to Lemma 32). By Lemma 32, we conclude $\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} v : \forall \alpha.s\sharp r_1 \preceq \alpha \to \mathsf{RGNHandle}\ \alpha \to \mathsf{RGN}\ \alpha\ \tau_a$. Let $e_{\text{wit}} = \Lambda\beta.\lambda k : \mathsf{RGN}\ s\sharp r_1\ \beta.\underline{\mathsf{witnessRGN}}\ s\sharp r_1\ s\sharp r_2\ [\beta]\ k$. Note that

$$\frac{\dfrac{\dfrac{\vdash_{\text{ctxt}} \cdot, \beta; \cdot; k : \mathsf{RGN}\ s\sharp r_1\ \beta; \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\}}{\cdot, \beta; \cdot, k : \mathsf{RGN}\ s\sharp r_1\ \beta; \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} k : \mathsf{RGN}\ s\sharp r_1\ \beta} \qquad \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{cast}} s\sharp r_1 \rightsquigarrow s\sharp r_2}{\cdot, \beta; \cdot, k : \mathsf{RGN}\ s\sharp r_1\ \beta; \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} \underline{\mathsf{witnessRGN}}\ s\sharp r_1\ s\sharp r_2\ [\beta]\ k : \mathsf{RGN}\ s\sharp r_2\ \beta}}{\cdot, \beta; \cdot; \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} \lambda k : \mathsf{RGN}\ s\sharp r_1\ \beta.\underline{\mathsf{witnessRGN}}\ s\sharp r_1\ s\sharp r_2\ [\beta]\ k : \mathsf{RGN}\ s\sharp r_1\ \beta \to \mathsf{RGN}\ s\sharp r_2\ \beta}}{\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} e_{\text{wit}} : s\sharp r_1 \preceq s\sharp r_2}$$

and

$$\frac{\dfrac{\dfrac{\dfrac{\cdot; \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{type}} s\sharp r_2 \qquad \cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} v : \forall \alpha.s\sharp r_1 \preceq \alpha \to \mathsf{RGNHandle}\ \alpha \to \mathsf{RGN}\ \alpha\ \tau_a}{\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} v\ [s\sharp r_2] : s\sharp r_1 \preceq s\sharp r_2 \to \mathsf{RGNHandle}\ s\sharp r_2 \to \mathsf{RGN}\ s\sharp r_2\ \tau_a} \qquad \cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} e_{\text{wit}} : s\sharp r_1 \preceq s\sharp r_2}{\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} v\ [s\sharp r_2]\ e_{\text{wit}} : \mathsf{RGNHandle}\ s\sharp r_2 \to \mathsf{RGN}\ s\sharp r_2\ \tau_a} \qquad \cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} \mathsf{handle}(s\sharp r_2) : \mathsf{RGNHandle}\ s\sharp r_2}{\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} v\ [s\sharp r_2]\ e_{\text{wit}}\ \mathsf{handle}(s\sharp r_2) : \mathsf{RGN}\ s\sharp r_2\ \tau_a}$$

Add $[T, s \mapsto S.r_2 \mapsto \{\}; v\ [s\sharp r_2]\ e_{\text{wit}}\ \mathsf{handle}(s\sharp r_2) \hookrightarrow ?]()$, which is well typed by $\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} v\ [s\sharp r_2]\ e_{\text{wit}}\ \mathsf{handle}(s\sharp r_2) : \mathsf{RGN}\ s\sharp r_2\ \tau_a$.

Case $([\tau_r \equiv s\sharp r_1], [r_1 \in dom(S)], [r_2 \notin dom(S)], [T, s \mapsto S.r_2 \mapsto \{\}; v\ [s\sharp r_2]\ e_{\text{wit}}\ \mathsf{handle}(s\sharp r_2) \hookrightarrow v'])$: Applying Preservation to (1a) $\vdash_{\text{tower}} T, s \mapsto S, r_2 \mapsto \{\} : \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\}$, (1b) $\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} v\ [s\sharp r_2]\ e_{\text{wit}}\ \mathsf{handle}(s\sharp r_2) : \mathsf{RGN}\ s\sharp r_2\ \tau_a$, and (1c) $T, s \mapsto S, r_2 \mapsto \{\}; v\ [s\sharp r_2]\ e_{\text{wit}}\ \mathsf{handle}(s\sharp r_2) \hookrightarrow v'$, we conclude $; \cdot; \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} v' : \mathsf{RGN}\ s\sharp r_2\ \tau_a$. By Lemma 38, we conclude $v' \equiv \kappa^{v'}$. Add $[v' \equiv \kappa^{v'}]$.

Case $([\tau_r \equiv s\sharp r_1], [r_1 \in dom(S)], [r_2 \notin dom(S)], [T, s \mapsto S.r_2 \mapsto \{\}; v\ [s\sharp r_2]\ e_{\text{wit}}\ \mathsf{handle}(s\sharp r_2) \hookrightarrow v'], [v' \equiv \kappa^{v'}])$: Add $[T, s \mapsto S, r_2 \mapsto \{\}; \kappa^{v'} \hookrightarrow_\kappa ?]()$, which is well typed by $;\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} \kappa^{v'} : \mathsf{RGN}\ s\sharp r_2\ \tau_a$.

Case $([\tau_r \equiv s\sharp r_1], [r_1 \in dom(S)], [r_2 \notin dom(S)], [T, s \mapsto S.r_2 \mapsto \{\}; v\ [s\sharp r_2]\ e_{\text{wit}}\ \mathsf{handle}(s\sharp r_2) \hookrightarrow v'], [v' \equiv \kappa^{v'}; T, s \mapsto S, r_2 \mapsto \{\}; \kappa^{v'} \hookrightarrow_\kappa S''; v'''])$: Applying Preservation to (2a) $\vdash_{\text{tower}} T, s \mapsto S, r_2 \mapsto \{\} : \mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\}$, (2b) $;\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}, r_2 \mapsto \{\} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}, r_2 \mapsto \{\} \vdash_{\text{exp}} \kappa^{v'} : \mathsf{RGN}\ s\sharp r_2\ \tau_a$, and (2c) $T, s \mapsto S, r_2 \mapsto \{\}; \kappa^v \hookrightarrow_\kappa S''; v'''$, we conclude that there exists $\overline{\mathcal{S}}'' \supseteq_{=\supseteq} \overline{\mathcal{S}}, r_2 \mapsto \{\}$ and $\mathcal{S}'' \supseteq_{=\supseteq} \mathcal{S}, r_2 \mapsto \{\}$ such that $\vdash_{\text{tower}} T, s \mapsto S'' : \mathcal{T}, s \mapsto \mathcal{S}'' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}''$ and $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S}'' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}'' \vdash_{\text{exp}} v''' : \tau_a$. Note that $\overline{\mathcal{S}}'' \supseteq_{=\supseteq} \overline{\mathcal{S}}, r_2 \mapsto \{\}$ implise $\overline{\mathcal{S}}'' \equiv \overline{\mathcal{S}}''', r_2 \mapsto \overline{\mathcal{R}}_2'''$. Note that $\mathcal{S}'' \supseteq_{=\supseteq} \mathcal{S}, r_2 \mapsto \{\}$ implise $\mathcal{S}'' \equiv \mathcal{S}''', r_2 \mapsto \mathcal{R}_2'''$. Note that $\vdash_{\text{tower}} T, s \mapsto S'' : \mathcal{T}, s \mapsto \mathcal{S}''', r_2 \mapsto \mathcal{R}_2''' : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}''', r_2 \mapsto \overline{\mathcal{R}}_2'''$ implies $S'' \equiv S''', r_2 \mapsto R'''$. Add $[S'' \equiv S''', r_2 \mapsto R_2''']$.

Case $([\tau_r \equiv s\sharp r_1], [r_1 \in dom(S)], [r_2 \notin dom(S)], [T, s \mapsto S.r_2 \mapsto \{\}; v\ [s\sharp r_2]\ e_{\text{wit}}\ \mathsf{handle}(s\sharp r_2) \hookrightarrow v'], [v' \equiv \kappa^{v'}; T, s \mapsto S, r_2 \mapsto \{\}; \kappa^{v'} \hookrightarrow_\kappa S''; v'''], [S'' \equiv S''', r_2 \mapsto R_2'''])$: Replace $\mathfrak{N}$ with $[T, s \mapsto S; \underline{\mathsf{letRGN}}\ [s\sharp r_1]\ [\tau_a]\ v \hookrightarrow_\kappa S'''[s\sharp \bullet/s\sharp r_2]; v'''[s\sharp \bullet/s\sharp r_2]]$.

Case $T, s \mapsto S; \underline{\mathsf{witnessRGN}}\ \sigma\sharp\rho_1\ \sigma\sharp\rho_2\ [\tau_a]\ v \hookrightarrow_\kappa ?$: Because $\mathfrak{D}$ is well typed, $T, s \mapsto S; \underline{\mathsf{witnessRGN}}\ \sigma\sharp\rho_1\ \sigma\sharp\rho_2\ [\tau_a]\ v \hookrightarrow_\kappa ?$ is well typed. Hence, there exists $\overline{\mathcal{T}}$, $\mathcal{T}$, $\overline{\mathcal{S}}$, $\mathcal{S}$, $r_2 \in dom(\mathcal{S})$, and τ_a such that $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$ and $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \underline{\mathsf{witnessRGN}}\ \sigma\sharp\rho_1\ \sigma\sharp\rho_2\ [\tau_a]\ v : \mathsf{RGN}\ s\sharp r_2\ \tau_a$ By inspection, the latter derivation must end with

$$\frac{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} v : \mathsf{RGN}\ s\sharp r_1\ \tau_a \qquad \dfrac{\overline{\mathcal{S}} = \overline{\mathcal{S}}_1, r_1 \mapsto \overline{\mathcal{R}}_1, \overline{\mathcal{S}}_2, r_2 \mapsto \overline{\mathcal{R}}_2, \overline{\mathcal{S}}_3}{\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{cast}} s\sharp r_1 \rightsquigarrow s\sharp r_2}}{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \underline{\mathsf{witnessRGN}}\ s\sharp r_1\ s\sharp r_2\ [\tau_a]\ v : \mathsf{RGN}\ s\sharp r_2\ \tau_a}$$

and $\sigma\sharp\rho_1 \equiv s\sharp r_1$, $\sigma\sharp\rho_2 \equiv s\sharp r_2$, $\overline{\mathcal{S}} \equiv \overline{\mathcal{S}}_1, r_1 \mapsto \overline{\mathcal{R}}_1, \overline{\mathcal{S}}_2, r_2 \mapsto \overline{\mathcal{R}}_2, \overline{\mathcal{S}}_3$. Consider the children of $T, s \mapsto S; \underline{\mathsf{witnessRGN}}\ \sigma\sharp\rho_1\ \sigma\sharp\rho_2\ [\tau_a]\ v \hookrightarrow_\kappa ?$:

Case $()$: Add $[\sigma\sharp\rho \equiv s\sharp r_1]$.

Case $([\sigma\sharp\rho_1 \equiv s\sharp r_1])$: Add $[\sigma'\sharp\rho \equiv s\sharp r_2]$.

Case $([\sigma\sharp\rho_1 \equiv s\sharp r_1], [\sigma\sharp\rho_2 \equiv s\sharp r_2])$: By Lemma 38, $v \equiv \kappa^v$. Add $[v \equiv \kappa^v]$.

Case $([\sigma\sharp\rho_1 \equiv s\sharp r_1], [\sigma\sharp\rho_2 \equiv s\sharp r_2], [v \equiv \kappa^v])$: Note that $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}_1, r_1 \mapsto \overline{\mathcal{R}}_1, \overline{\mathcal{S}}_2, r_2 \mapsto \overline{\mathcal{R}}_2, \overline{\mathcal{S}}_3$ implies $S = S_1, r_1 \mapsto R_1, S_2, r_2 \mapsto R_2, S_3$. Add $[S \equiv S_1, r_1 \mapsto R_1, S_2, r_2 \mapsto R_2, S_3]$.

Case $([\sigma\sharp\rho_1 \equiv s\sharp r_1], [\sigma\sharp\rho_2 \equiv s\sharp r_2], [v \equiv \kappa^v], [S \equiv S_1, r_1 \mapsto R_1, S_2, r_2 \mapsto R_2, S_3])$: Add $[T, s \mapsto S; \kappa^v \hookrightarrow_\kappa ?]()$, which is well typed by $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \kappa^v : \mathsf{RGN}\ s\sharp r_1\ \tau_a$.

Case $([\sigma\sharp\rho_1 \equiv s\sharp r_1], [\sigma\sharp\rho_2 \equiv s\sharp r_2], [v \equiv \kappa^v], [S \equiv S_1, r_1 \mapsto R_1, S_2, r_2 \mapsto R_2, S_3 \mid T, s \mapsto S; \kappa^v \hookrightarrow_\kappa S'; v'])$: Replace $\mathfrak{N}$ by $[T, s \mapsto S; \underline{\mathsf{witnessRGN}}\ s\sharp r_1\ s\sharp r_2\ [\tau_a]\ \kappa^v \hookrightarrow_\kappa S'; v']$.

Case $T, s \mapsto S; \underline{\mathsf{newRGNVar}}\ [\tau_r]\ [\tau_a]\ v_1\ v_2 \hookrightarrow_\kappa ?$: Because $\mathfrak{D}$ is well typed, $T, s \mapsto S; \underline{\mathsf{newRGNVar}}\ [\tau_r]\ [\tau_a]\ v_1\ v_2 \hookrightarrow_\kappa ?$ is well typed. Hence, there exists $\overline{\mathcal{T}}$, $\mathcal{T}$, $\overline{\mathcal{S}}$, $\mathcal{S}$, $r \in dom(\mathcal{S})$, and τ_a such that $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$ and . $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \underline{\mathsf{newRGNVar}}\ [\tau_r]\ [\tau_a]\ v_1\ v_2 : \mathsf{RGN}\ s\sharp r\ \tau_a$ By inspection, the latter derivation must end with

$$\frac{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} v_1 : \mathsf{RGNHandle}\ s\sharp r \qquad \cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} v_2 : \tau_a}{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \underline{\mathsf{newRGNVar}}\ [s\sharp r]\ [\tau_a]\ v_1\ v_2 : \mathsf{RGN}\ s\sharp r\ \tau_a}$$

and $\tau_r \equiv s\sharp r$. Consider the children of $T, s \mapsto S; \underline{\mathsf{newRGNVar}}\ [\tau_r]\ [\tau_a]\ v_1\ v_2 \hookrightarrow_\kappa ?$:

Case $()$: Add $[\tau_r \equiv s\sharp r]$.

Case $([\tau_r \equiv s\sharp r])$: By Lemma 38, $v_1 \equiv \mathsf{handle}(s\sharp r)$. Add $[v_1 \equiv \mathsf{handle}(s\sharp r)]$.

Case $([\tau_r \equiv s\sharp r], [v_1 \equiv \mathsf{handle}(s\sharp r)])$: Recall that $\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \mathsf{handle}(s\sharp r) : \mathsf{RGNHandle}\ s\sharp r$. By inspection, this derivation must end with

$$\frac{s \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}) \qquad r \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})(s))}{\cdot;\cdot;\mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \mathsf{handle}(s\sharp r) : \mathsf{RGNHandle}\ s\sharp r}$$

Hence, $r \in dom(\overline{\mathcal{S}})$. Note $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$ and $r \in dom(\overline{\mathcal{S}})$ implies $r \in dom(S)$. Add $[r \in dom(S)]$.

Case $([\tau_r \equiv s\sharp r], [v_1 \equiv \mathsf{handle}(s\sharp r)], [r \in dom(S)])$: Note that there exists $l \notin dom(S(r))$. Add $[l \notin dom(S(r))]$.

Case $([\tau_r \equiv s\sharp r], [v_1 \equiv \mathsf{handle}(s\sharp r)], [r \in dom(S)], [l \notin dom(S(r))])$: Replace $\mathfrak{N}$ with $[T, s \mapsto S; \underline{\mathsf{newRGNVar}}\ [s\sharp r]\ [\tau_a]\ \mathsf{handle}(s\sharp r)\ v_2 \hookrightarrow_\kappa S\{(r, l) \mapsto w\}; \langle l \rangle_{s\sharp r}]$.

Case $T, s \mapsto S; \underline{\mathsf{readRGNVar}}\ [\tau_r]\ [\tau_a]\ v \hookrightarrow_\kappa ?$: Because $\mathfrak{D}$ is well typed, $T, s \mapsto S; \underline{\mathsf{readRGNVar}}\ [\tau_r]\ [\tau_a]\ v \hookrightarrow_\kappa ?$ is well typed. Hence, there exists $\overline{\mathcal{T}}$, $\mathcal{T}$, $\overline{\mathcal{S}}$, $\mathcal{S}$, $r \in dom(\mathcal{S})$, and τ_a such that $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$ and $\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \underline{\mathsf{readRGNVar}}\ [\tau_r]\ [\tau_a]\ v : \mathsf{RGN}\ s\sharp r\ \tau_a$ By inspection, the latter derivation must end with

$$\frac{\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} v : \mathsf{RGNVar}\ s\sharp r\ \tau_a}{\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \underline{\mathsf{readRGNVar}}\ [s\sharp r]\ [\tau_a]\ v : \mathsf{RGN}\ s\sharp r\ \tau_a}$$

and $\tau_r \equiv s\sharp r$. Consider the children of $T, s \mapsto S; \underline{\mathsf{readRGNVar}}\ [\tau_r]\ [\tau_a]\ v \hookrightarrow_\kappa ?$:

Case $()$: Add $[\tau_r \equiv s\sharp r]$.

Case $([\tau_r \equiv s\sharp r])$: By Lemma 38, $v \equiv \langle l \rangle_{s\sharp r}$. Add $[v \equiv \langle l \rangle_{s\sharp r}]$.

Case $([\tau_r \equiv s\sharp r], [v \equiv \langle l \rangle_{s\sharp r}])$: Recall that $\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \langle l \rangle_{s\sharp r} : \mathsf{RGNVar}\ s\sharp r\ \tau_a$. By inspection, this derivation must end with

$$\frac{\begin{array}{c}\vdash_{\text{ctxt}} \cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \\ s \in dom(\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}) \qquad r \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})(s)) \qquad l \in dom((\overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}})(s, r)) \qquad \tau_a = (\mathcal{T}, s \mapsto \mathcal{S})(s, r, l)\end{array}}{\cdot; \cdot; \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}} \vdash_{\text{exp}} \langle l \rangle_{s\sharp r} : \mathsf{RGNVar}\ s\sharp r\ \tau_a}$$

Hence, $r \in dom(\overline{\mathcal{S}})$ and $l \in dom(\overline{\mathcal{S}}(r))$. Note $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$ and $r \in dom(\overline{\mathcal{S}})$ implies $r \in dom(S)$. Add $[r \in dom(S)]$.

Case $([\tau_r \equiv s\sharp r], [v \equiv \langle l \rangle_{s\sharp r}], [r \in dom(S)])$: Note $\vdash_{\text{tower}} T, s \mapsto S : \mathcal{T}, s \mapsto \mathcal{S} : \overline{\mathcal{T}}, s \mapsto \overline{\mathcal{S}}$ and $l \in dom(\overline{\mathcal{S}}(r))$ implies $l \in dom(S(r))$. Add $[l \in dom(S(s))]$.

Case $([\tau_r \equiv s\sharp r], [v \equiv \langle l \rangle_{s\sharp r}], [r \in dom(S)], [l \in dom(S(r))])$: Note $l \in dom(S(r))$ implies $v' = S(r, l)$. Replace $\mathfrak{N}$ with $[T, s \mapsto S; \underline{\mathsf{readRGNVar}}\ [s\sharp r]\ [\tau_a]\ \langle l \rangle_{s\sharp r} \hookrightarrow_\kappa S; v']$.

□

3.5.4 Soundness

Theorem 4 (Soundness)
If $\cdot; \cdot; \cdot; \cdot : \cdot \vdash_{\text{exp}} e : \tau$, then any execution of e (in $\cdot$) either terminates with a value v (such that $\cdot; \cdot; \cdot : \cdot \vdash_{\text{exp}} v : \tau$) or diverges.

Proof. Let $[\cdot; e \hookrightarrow ?]() \longrightarrow \mathfrak{D}_1 \longrightarrow \mathfrak{D}_2 \longrightarrow \cdots$ be an execution of e. Note that $[\cdot; e \hookrightarrow ?]()$ is well-typed by $\vdash_{\text{tower}} \cdot : \cdot : \cdot$ and $\cdot; \cdot; \cdot; \cdot : \cdot \vdash_{\text{exp}} e : \tau$. By Progress, every $\mathfrak{D}_i$ is well typed;

(1) Suppose that for all $\mathfrak{D}_n$ such that $[T; e \hookrightarrow ?]() \longrightarrow^* \mathfrak{D}_n$, there exists $\mathfrak{D}_{n+1}$ such that $\mathfrak{D}_n \longrightarrow \mathfrak{D}_{n+1}$. Then, e diverges.

(2) Suppose that there exists $\mathfrak{D}_n$ such that $[T; e \hookrightarrow ?]() \longrightarrow^* \mathfrak{D}_n$, such tha tthere does not exist $\mathfrak{D}_{n+1}$ such that $\mathfrak{D}_n \longrightarrow \mathfrak{D}_{n+1}$.

(a) Suppose $\mathfrak{D}_n$ contains no pending judgements. By Lemma 21, $\mathfrak{D}_n \equiv [T; e \hookrightarrow v]$. Then, e terminates with the value v. By Preservation, $\cdot; \cdot; \cdot : \cdot \vdash_{\text{exp}} v : \tau$.

(b) Suppose $\mathfrak{D}_n$ contains pending judgements. By Progress, there exists $\mathfrak{D}'$ such that $\mathfrak{D}_n \longrightarrow \mathfrak{D}'$, contradicting the assumption that there does not exist $\mathfrak{D}_{n+1}$ such that $\mathfrak{D}_n \longrightarrow \mathfrak{D}_{n+1}$. Thus, e cannot get stuck.

□

$$\begin{array}{lrcl} \text{Surface commands} & \kappa^v & ::= & \ldots \mid \underline{\mathsf{writeRGNVar}}\ [\tau_r]\ [\tau_a]\ v_1\ v_2 \\ \text{Computation commands} & \kappa^v & ::= & \ldots \mid \underline{\mathsf{writeRGNVar}}\ [\tau_r]\ [\tau_a]\ v_1\ v_2 \end{array}$$

$\boxed{T, s \mapsto S; \kappa^v \hookrightarrow_\kappa S'; v}$

$$\frac{\begin{array}{cc} \tau_r \equiv s\sharp r & v_1 \equiv \langle l \rangle_{s\sharp r} \\ r \in dom(S) & l \in dom(S(r)) \end{array}}{S; \underline{\mathsf{readRGNVar}}\ [\tau_r]\ [\tau_a]\ v_1\ v_2 \hookrightarrow_\kappa S\{(r,l) \mapsto v_2\}; ()}$$

$\boxed{\Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e : \tau}$

$$\frac{\Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v_1 : \mathsf{RGNVar}\ \tau_r\ \tau_a \Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v_2 : \tau_a}{\Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \underline{\mathsf{writeRGNVar}}\ [\tau_r]\ [\tau_a]\ v_1\ v_2 : 1}$$

Figure 32: Extensions to $\mathsf{F}^{\mathsf{RGN}}$ for $\underline{\mathsf{writeRGNVar}}$

3.6 Extensions

We consider two easy extensions to the language $\mathsf{F}^{\mathsf{RGN}}$: mutable variables and fixed-point variables.

3.6.1 Mutable variables

Figure 32 presents the extensions necessary to support mutable variables The command $\underline{\mathsf{writeRGNVar}}\ [\tau_r]\ [\tau_a]\ v_1\ v_2$ overwrites the value stored at the location v_1 with the value v_2. All lemmas and theorems can be extended to support mutable variables in a straight-forward manner. In the special case where programs do not contain $\underline{\mathsf{letRGN}}$, we obtain an alternative proof for the soundness of strict monadic state [23].

3.6.2 Fixed-point variables

Figure 33 presents the extensions necessary to support fixed-point variables. The command $\underline{\mathsf{fixRGNVar}}\ [\tau_r]\ [\tau_a]\ v_1\ v_2$ allocates a value of type τ_a in the region indexed by τ_r; the value is produced by the function v_2 which is applied to the location where the allocated value is to be stored. This provides a means of allocating recursive functions. (Recursive structures could be accomodated with the addition of recursive types.)

The dynamic semantics for $\underline{\mathsf{fixRGNVar}}$ make use of a dummy storable value $\Diamond$. The typing rule for $\Diamond$ admits arbitrary well-formed types. (We slightly abuse notation here; $\Diamond$ is not technically an expression form. A $\Diamond$ can only appear in the range of a region.) However, $\Diamond$ is not a value and cannot be the result of any computation. It serves as a place holder in the store, marking the location where the recursive knot will be tied. It also ensures that the tower is well-formed with respect to the extended tower type necessary to prove that $v_2\ \langle l \rangle_{s\sharp r}$ is well-typed.

We note that the typing rule for $\underline{\mathsf{fixRGNVar}}$ requires that the function v_2 has the type $\mathsf{RGNVar}\ \tau_r\ \tau_a \to \tau_a$. This is a pure function, not a monadic computation. Hence, it is safe to evaluate with the location bound to $\Diamond$ (where the allocated value is to be stored), because no computation (and, hence, no reading of region allocated values) can occur during the evaluation of the application of v_2 to the location. On the other hand, v_2 can return a (suspended) computation that reads the allocated value, since this computation cannot occur until after the knot has been tied. For example, the following expression[2] returns a location of type

[2]In which we assume h has type $\mathsf{RGNHandle}\ \tau_r$ and we introduce the following notation:

$$\begin{array}{l} \mathsf{bind}\ f : \tau_a \Leftarrow e_1; e_2 \equiv \mathsf{let}\ k = e_1\ \mathsf{in} \\ \qquad\qquad \underline{\mathsf{thenRGN}}\ [\tau_r]\ [\tau_a]\ [\tau_b]\ k\ (\lambda f : \tau_a . e_2) \end{array}$$

$$
\begin{array}{lrcl}
\text{Surface commands} & \kappa^v & ::= & \ldots \mid \underline{\mathsf{fixRGNVar}}\ [\tau_r]\ [\tau_a]\ v_1\ v_2 \\
\text{Computation commands} & \kappa^v & ::= & \ldots \mid \underline{\mathsf{fixRGNVar}}\ [\tau_r]\ [\tau_a]\ v_1\ v_2 \\
\text{Storable values} & w & ::= & v \mid \Diamond \\
\text{Regions} & R & ::= & \{l_1 \mapsto w_1, \ldots, l_n \mapsto w_n\}
\end{array}
$$

$\boxed{T, s \mapsto S; \kappa^v \hookrightarrow_\kappa S'; v}$

$$
\frac{\begin{array}{c}\tau_r \equiv s\sharp r \qquad v_1 \equiv \mathsf{handle}(s\sharp r) \\ r \in dom(S) \qquad l \notin dom(S(r)) \\ T, s \mapsto S\{(r,l) \mapsto \Diamond\}; v_2\ \langle l\rangle_{s\sharp r} \hookrightarrow v'\end{array}}{S; \underline{\mathsf{fixRGNVar}}\ [\tau_r]\ [\tau_a]\ v_1\ v_2 \hookrightarrow S\{(r,l) \mapsto v'\}, \langle l\rangle_{s\sharp r}}
$$

$\boxed{\Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e : \tau}$

$$
\frac{\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \qquad \Delta; \overline{\mathcal{T}} \vdash_{\text{type}} \tau}{\Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \Diamond : \tau}
$$

$$
\frac{\Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v_1 : \mathsf{RGNHandle}\ \tau_r \qquad \Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} v_2 : \mathsf{RGNVar}\ \tau_r\ \tau_a \to \tau_a}{\Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \underline{\mathsf{fixRGNVar}}\ [\tau_r]\ [\tau_a]\ v_1\ v_2 : \mathsf{RGN}\ \tau_r\ (\mathsf{RGNVar}\ \tau_r\ \tau_a)}
$$

Figure 33: Extensions to $\mathsf{F}^{\mathsf{RGN}}$ for $\underline{\mathsf{fixRGNVar}}$

$\mathsf{RGNVar}\ \tau_r\ (\mathsf{int} \to \mathsf{RGN}\ \tau_r\ \mathsf{int})$, which points to a (monadic) function that evaluates factorials:

$$
\begin{array}{l}
\underline{\mathsf{fixRGNVar}}\ [\tau_r]\ [\mathsf{int} \to \mathsf{RGN}\ \tau_r\ \mathsf{int}]\ h \\
\quad (\lambda f : \mathsf{RGNVar}\ \tau_r\ (\mathsf{int} \to \mathsf{RGN}\ \tau_r\ \mathsf{int}). \\
\quad\ \lambda n : \mathsf{int}. \\
\quad\ \mathsf{if}\ n = 0\ \mathsf{then}\ \underline{\mathsf{returnRGN}}\ \tau_r\ 1 \\
\quad\quad \mathsf{else\ bind}\ g : \mathsf{int} \to \mathsf{RGN}\ \tau_r\ \mathsf{int} \Leftarrow \underline{\mathsf{readRGNVar}}\ [\tau_r]\ [\mathsf{int} \to \mathsf{RGN}\ \tau_r\ \mathsf{int}]\ f; \\
\quad\quad\quad \mathsf{bind}\ m : \mathsf{int} \Leftarrow g\ (n-1); \\
\quad\quad\quad \underline{\mathsf{returnRGN}}\ [\tau_r]\ [\mathsf{int}]\ (n * m)
\end{array}
$$

where k is fresh and τ_r and τ_b are inferred from context

Translations yielding region domains

Region domains

$$\mathbb{T}^{\vdash_{\text{rdom}}}_{\overline{\mathcal{R}}}\left[\!\!\left[\frac{i \neq j \Rightarrow l_i \neq l_j \;^{i\in 1\ldots n, j\in 1\ldots n}}{\vdash_{\text{rdom}} \{l_1,\ldots,l_n\}}\right]\!\!\right] \quad = \quad \{l_1,\ldots,l_n\}$$

Translations yielding stack domains

Stack domains

$$\mathbb{T}^{\vdash_{\text{sdom}}}_{\overline{\mathcal{S}}}\left[\!\!\left[\frac{}{\vdash_{\text{sdom}} \cdot}\right]\!\!\right] \quad = \quad \cdot$$

$$\mathbb{T}^{\vdash_{\text{sdom}}}_{\overline{\mathcal{S}}}\left[\!\!\left[\frac{\vdash_{\text{sdom}} \overline{\mathcal{S}} \qquad r \notin dom(\overline{\mathcal{S}}) \qquad \vdash_{\text{rdom}} \overline{\mathcal{R}}}{\vdash_{\text{sdom}} \overline{\mathcal{S}}, r \mapsto \overline{\mathcal{R}}}\right]\!\!\right] \quad = \quad \mathbb{T}^{\vdash_{\text{sdom}}}_{\overline{\mathcal{S}}}\left[\!\!\left[\overline{\mathcal{S}}\right]\!\!\right], r \mapsto \mathbb{T}^{\vdash_{\text{rdom}}}_{\overline{\mathcal{R}}}\left[\!\!\left[\overline{\mathcal{R}}\right]\!\!\right]$$

Translations yielding tower domains

Stack domains

$$\mathbb{T}^{\vdash_{\text{sdom}}}_{\overline{\mathcal{T}}}\left[\!\!\left[\vdash_{\text{sdom}} \overline{\mathcal{S}}\right]\!\!\right] \quad = \quad \begin{cases} \cdot & \text{if } \overline{\mathcal{S}} = \cdot \\ \cdot, s \mapsto \mathbb{T}^{\vdash_{\text{sdom}}}_{\overline{\mathcal{S}}}\left[\!\!\left[\overline{\mathcal{S}}\right]\!\!\right] & \text{otherwise} \end{cases}$$

Figure 34: Translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$ (Stacks (I))

4 The Translation

In this section we present a type- and semantics-preserving translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$. Many of the key components of the translation should be obvious from the suggestive naming of the previous sections. We clearly intend letregion to be translated (in some fashion) to letRGN. Likewise, we can expect types of the form (μ, ρ) to be translated to types of the form RGNVar τ_r τ_a. It further seems likely that the outlives relation $\rho \succeq \rho'$ should be related to the witness functions $\tau'_r \preceq \tau_r$. We present the translation in stages, as there are some subtleties that require explanation.

We start with a few preliminaries. We assume injections from the sets $RVars^{SEC}$ and $Vars^{SEC}$ to the sets $TVars^{FRGN}$ and $Vars^{FRGN}$ respectively. In the translation, applications of such injections will be clear from context and we freely use variables from source objects in target objects. We further assume two additional injections from the set $RVars^{SEC}$ to the set $Vars^{FRGN}$; the first, written h_ϱ will denote the handle for the region ϱ, while the second, written w_ϱ will denote the witnesses for the region ϱ.

The translation is a typed call-by-value monad translation, similar to the standard translation given by Sabry and Wadler [22]. We have not attempted to optimize the translation to avoid the introduction of "administrative" redexes. We feel that this simplifies the translation, and it does not significantly complicate the proof that the translation preserves the semantics, owing to the fact that only three expression forms in the source calculus are value forms. The translation is given by a number of functions: $\mathbb{T}^{src}_{dst}[\![\cdot]\!]$ translates from the syntactic class *src* in the Single Effect Calculus to the syntactic class *dst* in $\mathsf{F}^{\mathsf{RGN}}$.

Figure 38 shows the translation of types. As expected, the type (μ, ρ) is translated to RGNVar $\mathbb{T}^{\vdash_{\text{place}}}_{\tau}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho\right]\!\!\right]$ $\mathbb{T}^{\vdash_{\text{btype}}}_{\tau}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \mu\right]\!\!\right]$, whereby region allocated values in the source are also region allocated in the target. The translations of primitive types and product types are trivial. More interesting are the translations of function types and region abstraction types. Functions with effects bounded by the region θ are translated into pure functions that yield computations taking place in stack of regions with θ as a member. Region abstractions are translated into type abstractions. Because the target calculus requires explicit region handles for allocation, each time a region is in scope in the source calculus, the region handle must be in scope in the target calculus. This explains the appearance of the RGNHandle ϱ type in the translation. Likewise, the target calculus makes witness functions explicit, whereas in the source calculus such coercions are implied by $\succeq$ related regions. Hence, we interpret $\varrho \succeq \{\rho_1,\ldots,\rho_n\}$ as an n-

Translations yielding type contexts

Region contexts

$$\mathbb{T}^{\vdash_{\text{rctxt}}}_{\Delta}\left[\!\left[\frac{\vdash_{\text{sdom}} \overline{\mathcal{S}}}{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \cdot}\right]\!\right] = \cdot$$

$$\mathbb{T}^{\vdash_{\text{rctxt}}}_{\Delta}\left[\!\left[\frac{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta \qquad \varrho \notin dom(\Delta) \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{eff}} \varphi}{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta, \varrho \succeq \varphi}\right]\!\right] = \mathbb{T}^{\vdash_{\text{rctxt}}}_{\Delta}\left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right], \varrho$$

Figure 35: Translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$ (Region contexts (I))

Translations yielding types

Places

$$\mathbb{T}^{\vdash_{\text{place}}}_{\tau}\left[\!\left[\frac{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta \qquad \varrho \in dom(\Delta)}{\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \varrho}\right]\!\right] = \varrho$$

$$\mathbb{T}^{\vdash_{\text{place}}}_{\tau}\left[\!\left[\frac{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta \qquad r \in dom(\overline{\mathcal{S}})}{\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} r}\right]\!\right] = s\sharp r$$

$$\mathbb{T}^{\vdash_{\text{place}}}_{\tau}\left[\!\left[\frac{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta}{\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \bullet}\right]\!\right] = \begin{cases} \circ\sharp\bullet & \text{if } \overline{\mathcal{S}} = \cdot \\ s\sharp\bullet & \text{otherwise} \end{cases}$$

Figure 36: Translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$ (Places (I))

Translations yielding value contexts

Region contexts

$$\mathbb{T}^{\vdash_{\text{rctxt}}}_{\Gamma}\left[\!\left[\frac{\vdash_{\text{sdom}} \overline{\mathcal{S}}}{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \cdot}\right]\!\right] = \cdot$$

$$\mathbb{T}^{\vdash_{\text{rctxt}}}_{\Gamma}\left[\!\left[\frac{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta \qquad \varrho \notin dom(\Delta) \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{eff}} \varphi}{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta, \varrho \succeq \varphi}\right]\!\right] =$$

$$\mathbb{T}^{\vdash_{\text{rctxt}}}_{\Gamma}\left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right], h_{\varrho} : \mathsf{RGNHandle}\ \varrho, w_{\varrho} : (\mathbb{T}^{\vdash_{\text{place}}}_{\tau}\left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho_1\right]\!\right] \preceq \varrho \times \cdots \times \mathbb{T}^{\vdash_{\text{place}}}_{\tau}\left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho_n\right]\!\right] \preceq \varrho)$$

where $\varphi = \{\rho_1, \ldots, \rho_n\}$

Figure 37: Translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$ (Region contexts (II))

Translations yielding types

Types

$$\mathbb{T}_\tau^{\vdash \text{type}} \left[\!\!\left[\frac{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta}{\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \mathsf{bool}} \right]\!\!\right] = \mathsf{bool}$$

$$\mathbb{T}_\tau^{\vdash \text{type}} \left[\!\!\left[\frac{\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \mu \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho}{\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} (\mu, \rho)} \right]\!\!\right] = \mathsf{RGNVar}\ \mathbb{T}_\tau^{\vdash \text{place}} [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho]\!]\ \mathbb{T}_\tau^{\vdash \text{btype}} [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \mu]\!]$$

Boxed types

$$\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\frac{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta}{\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}} \right]\!\!\right] = \mathsf{int}$$

$$\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\frac{\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_1 \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \theta \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_2}{\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \tau_1 \xrightarrow{\theta} \tau_2} \right]\!\!\right] =$$
$$\mathbb{T}_\tau^{\vdash \text{type}} [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_1]\!] \to \mathsf{RGN}\ \mathbb{T}_\tau^{\vdash \text{place}} [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \theta]\!]\ \mathbb{T}_\tau^{\vdash \text{type}} [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_2]\!]$$

$$\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\frac{\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_1 \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_2}{\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \tau_1 \times \tau_2} \right]\!\!\right] = \mathbb{T}_\tau^{\vdash \text{type}} [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_1]\!] \times \mathbb{T}_\tau^{\vdash \text{type}} [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_2]\!]$$

$$\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\frac{\Delta; \overline{\mathcal{S}} \vdash_{\text{eff}} \varphi \qquad \Delta, \varrho \succeq \varphi; \overline{\mathcal{S}} \vdash_{\text{place}} \theta \qquad \Delta, \varrho \succeq \varphi; \overline{\mathcal{S}} \vdash_{\text{type}} \tau}{\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \Pi \varrho \succeq \varphi.^{\theta} \tau} \right]\!\!\right] =$$
$$\forall \varrho. (\mathbb{T}_\tau^{\vdash \text{place}} [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho_1]\!] \preceq \varrho \times \cdots \times \mathbb{T}_\tau^{\vdash \text{place}} [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho_n]\!] \preceq \varrho) \to$$
$$\mathsf{RGNHandle}\ \varrho \to$$
$$\mathsf{RGN}\ \mathbb{T}_\tau^{\vdash \text{place}} [\![\Delta, \varrho \succeq \varphi; \overline{\mathcal{S}} \vdash_{\text{place}} \theta]\!]\ \mathbb{T}_\tau^{\vdash \text{type}} [\![\Delta, \varrho \succeq \varphi; \overline{\mathcal{S}} \vdash_{\text{type}} \tau]\!]$$
$$\text{where } \varphi = \{\rho_1, \ldots, \rho_n\}$$

Figure 38: Translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$ (Types)

Translations yielding stack types

Stacks types

$$\mathbb{T}_{\mathcal{S}}^{\vdash \text{stype}} \left[\!\!\left[\frac{\begin{array}{c} \vdash_{\text{sdom}} \overline{\mathcal{S}} \\ dom(\overline{\mathcal{S}}) = dom(\mathcal{S}) \\ \forall r \in dom(\overline{\mathcal{S}}).\ dom(\overline{\mathcal{S}}(r)) = dom(\mathcal{S}(r)) \\ \forall r \in dom(\overline{\mathcal{S}}). \forall l \in dom(\overline{\mathcal{S}}(r)).\ \cdot; \overline{\mathcal{S}} \vdash_{\text{btype}} \mathcal{S}(r, l) \end{array}}{\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}} \right]\!\!\right] = \mathcal{S}^*$$

where
$$dom(S) = dom(S^*)$$
$$\forall r \in dom(S). dom(S(r)) = dom(S^*(r))$$
$$\forall r \in dom(S). \forall l \in dom(\overline{\mathcal{S}}(r)). S^*(r, l) = \mathbb{T}_\tau^{\vdash \text{btype}} [\![\cdot; \overline{\mathcal{S}} \vdash_{\text{btype}} \mathcal{S}(r, l)]\!]$$

Translations yielding tower types

Stack types

$$\mathbb{T}_{\mathcal{T}}^{\vdash \text{stype}} [\![\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}]\!] = \begin{cases} \cdot & \text{if } \mathcal{S} = \cdot \\ \cdot, s \mapsto \mathbb{T}_{\mathcal{S}}^{\vdash \text{stype}} [\![\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}]\!] & \text{otherwise} \end{cases}$$

Figure 39: Translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$ (Stacks (II))

Translations yielding value contexts

Value contexts

$$\mathbb{T}^{\vdash_{\text{vctxt}}}_{\Gamma}\left[\!\!\left[\frac{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta}{\Delta; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \cdot}\right]\!\!\right] \;=\; \cdot$$

$$\mathbb{T}^{\vdash_{\text{vctxt}}}_{\Gamma}\left[\!\!\left[\frac{\Delta; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \Gamma \qquad x \notin dom(\Gamma) \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau}{\Delta; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \Gamma, x : \tau}\right]\!\!\right] \;=\; \mathbb{T}^{\vdash_{\text{vctxt}}}_{\Gamma}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \Gamma\right]\!\!\right], x : \mathbb{T}^{\vdash_{\text{type}}}_{\tau}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau\right]\!\!\right]$$

Figure 40: Translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$ (Value contexts)

Translations yielding types

Witnesses

$$\mathbb{T}^{\vdash_{\text{rr}}}_{\tau}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho'\right]\!\!\right] \;=\; \mathbb{T}^{\vdash_{\text{place}}}_{\tau}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho'\right]\!\!\right] \preceq \mathbb{T}^{\vdash_{\text{place}}}_{\tau}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho\right]\!\!\right]$$

$$\mathbb{T}^{\vdash_{\text{re}}}_{\tau}\left[\!\!\left[\frac{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta \quad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho_i \;{}^{i \in 1 \ldots n}}{\Delta; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \{\rho_1, \ldots, \rho_n\}}\right]\!\!\right] \;=\; (\mathbb{T}^{\vdash_{\text{rr}}}_{\tau}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho_1\right]\!\!\right] \times \cdots \times \mathbb{T}^{\vdash_{\text{rr}}}_{\tau}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho_n\right]\!\!\right])$$

Translations yielding terms

Witnesses

$$\mathbb{T}^{\vdash_{\text{rr}}}_{v}\left[\!\!\left[\frac{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta \quad \Delta(\varrho) = \{\rho_1, \ldots, \rho_i, \ldots, \rho_n\}}{\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \varrho \succeq \rho_i}\right]\!\!\right] \;=\; \Lambda\beta.\lambda k : \mathsf{RGN}\ \mathbb{T}^{\vdash_{\text{place}}}_{\tau}\left[\!\!\left[\Delta \overline{\mathcal{S}} \vdash_{\text{place}} \rho_i\right]\!\!\right]\ \beta.\mathsf{let}\ w = \mathsf{sel}_i\ w_{\varrho}\ \mathsf{in}\ w\ [\beta]\ k$$

$$\mathbb{T}^{\vdash_{\text{rr}}}_{v}\left[\!\!\left[\frac{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta \quad \overline{\mathcal{S}} = \overline{\mathcal{S}}_1, r_1 \mapsto \overline{\mathcal{R}}_1, \overline{\mathcal{S}}_2, r_2 \mapsto \overline{\mathcal{R}}_2, \overline{\mathcal{S}}_3}{\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} r_2 \succeq r_1}\right]\!\!\right] \;=\; \Lambda\beta.\lambda k : \mathsf{RGN}\ s\sharp r_1\ \beta.\underline{\mathsf{witnessRGN}}\ s\sharp r_1\ s\sharp r_2\ [\beta]\ k$$

$$\mathbb{T}^{\vdash_{\text{rr}}}_{v}\left[\!\!\left[\frac{\Delta; \overline{\mathcal{S}} \vdash_{\text{re}} r \succeq \{r_1, \ldots, r_n\}}{\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} r \succeq r_i}\right]\!\!\right] =$$

$$\Lambda\beta.\lambda k : \mathsf{RGN}\ \mathbb{T}^{\vdash_{\text{place}}}_{\tau}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} r_i\right]\!\!\right]\ \beta.\mathsf{let}\ w = \mathsf{sel}_i\ \mathbb{T}^{\vdash_{\text{re}}}_{v}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{re}} r \succeq \{r_1, \ldots, r_n\}\right]\!\!\right]\ \mathsf{in}\ w\ [\beta]\ k$$

$$\mathbb{T}^{\vdash_{\text{rr}}}_{v}\left[\!\!\left[\frac{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta \qquad \overline{\mathcal{S}} = \overline{\mathcal{S}}_1, r_1 \mapsto R_1, \overline{\mathcal{S}}_2}{\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \bullet \succeq r_1}\right]\!\!\right] \;=\; \Lambda\beta.\lambda k : \mathsf{RGN}\ s\sharp r_1\ \beta.\underline{\mathsf{witnessRGN}}\ s\sharp r_1\ s\sharp \bullet\ [\beta]\ k$$

$$\mathbb{T}^{\vdash_{\text{rr}}}_{v}\left[\!\!\left[\frac{\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho}{\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho}\right]\!\!\right] \;=\; \Lambda\beta.\lambda k : \mathsf{RGN}\ \mathbb{T}^{\vdash_{\text{place}}}_{\tau}\left[\!\!\left[\Delta \overline{\mathcal{S}} \vdash_{\text{place}} \rho\right]\!\!\right]\ \beta.k$$

$$\mathbb{T}^{\vdash_{\text{rr}}}_{v}\left[\!\!\left[\frac{\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho' \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho' \succeq \rho''}{\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho''}\right]\!\!\right] =$$

$$\Lambda\beta.\lambda k : \mathsf{RGN}\ \mathbb{T}^{\vdash_{\text{place}}}_{\tau}\left[\!\!\left[\Delta \overline{\mathcal{S}} \vdash_{\text{place}} \rho''\right]\!\!\right]\ \beta.\mathsf{let}\ k' = \mathbb{T}^{\vdash_{\text{rr}}}_{v}\left[\!\!\left[\Delta \overline{\mathcal{S}} \vdash_{\text{rr}} \rho' \succeq \rho''\right]\!\!\right]\ [\beta]\ k\ \mathsf{in}\ \mathbb{T}^{\vdash_{\text{rr}}}_{v}\left[\!\!\left[\Delta \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho'\right]\!\!\right]\ [\beta]\ k'$$

$$\mathbb{T}^{\vdash_{\text{re}}}_{v}\left[\!\!\left[\frac{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta \quad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho_i \;{}^{i \in 1 \ldots n}}{\Delta; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \{\rho_1, \ldots, \rho_n\}}\right]\!\!\right] \;=\; (\mathbb{T}^{\vdash_{\text{rr}}}_{v}\left[\!\!\left[\Delta \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho_1\right]\!\!\right], \ldots, \mathbb{T}^{\vdash_{\text{rr}}}_{v}\left[\!\!\left[\Delta \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho_n\right]\!\!\right])$$

Figure 41: Translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$ (witnesses)

Translations yielding terms

Places

$$\mathbb{T}_v^{\vdash_{\text{place}}}\left[\!\left[\frac{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta \qquad \varrho \in dom(\Delta)}{\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \varrho}\right]\!\right] = h_\varrho$$

$$\mathbb{T}_v^{\vdash_{\text{place}}}\left[\!\left[\frac{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta \qquad r \in dom(\overline{\mathcal{S}})}{\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} r}\right]\!\right] = \mathsf{handle}(s\sharp r)$$

$$\mathbb{T}_v^{\vdash_{\text{place}}}\left[\!\left[\frac{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta}{\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \bullet}\right]\!\right] = \begin{cases} \mathsf{handle}(\circ\sharp\bullet) & \text{if } \overline{\mathcal{S}} = \cdot \\ \mathsf{handle}(s\sharp\bullet) & \text{otherwise} \end{cases}$$

Figure 42: Translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$ (Places (II))

tuple of functions, each witnessing a coercion from region ρ_i to ϱ. This interpretation is formalized by the $\mathbb{T}_\tau^{\vdash_{\text{place}}}\left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho_i\right]\!\right] \preceq \varrho$ types.[3]

We extend the type translation to contexts in the obvious way. In addition to translating region variables to type variables and translating the types of variables in value contexts, we have additional translations from region contexts to value contexts. As explained above, region handles and witness functions are explicit values in the target calculus. Hence, our translation maintains the invariant that whenever a region variable $\varrho \succeq \{\rho_1, \ldots, \rho_n\}$ is in scope in the source calculus, the variables ϱ^h and ϱ^w are in scope in the target calculus. The variable ϱ^h (of type $\mathsf{RGNHandle}\ \varrho$) is the handle for the region ϱ and the variable ϱ^w (of type $\mathbb{T}_\tau^{\vdash_{\text{place}}}\left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho_1\right]\!\right] \preceq \varrho \times \cdots \times \mathbb{T}_\tau^{\vdash_{\text{place}}}\left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho_n\right]\!\right] \preceq \varrho$) is the tuple holding the witness functions that coerce to region ϱ.

Figure 41 shows the translation of witness terms. The first six translations in the second set of translations map the reflexive, transitive closure of the syntactic constraints in the source Δ and $\overline{\mathcal{S}}$ into an appropriate coercion function. The final translation collects a set of coercion functions into a tuple; such a term is suitable as an argument to the translation of a region abstraction.

Figures 43, 44, 45, and 46 shows the translation of terms. In order to make the translation easier to read, we introduce the following notation:

$$\begin{aligned} \mathsf{bind}\ f : \tau_a \Leftarrow e_1; e_2 \equiv\ & \mathsf{let}\ k = e_1\ \mathsf{in} \\ & \underline{\mathsf{thenRGN}}\ [\tau_r]\ [\tau_a]\ [\tau_b]\ k\ (\lambda f : \tau_a.e_2) \\ & \quad \text{where } k \text{ fresh} \end{aligned}$$

where τ_r and τ_b are inferred from context. Note that this induces the following derived rules:

$$\frac{T; e_1 \hookrightarrow v}{T; \mathsf{bind}\ f : \tau_a \Leftarrow e_1; e_2 \hookrightarrow \underline{\mathsf{thenRGN}}\ [\tau_r]\ [\tau_a]\ [\tau_b]\ v\ (\lambda f : \tau_a.e_2)}$$

$$\frac{\Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_1 : \mathsf{RGN}\ \tau_r\ \tau_a \qquad \Delta; \Gamma, f : \tau_a; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} e_2 : \mathsf{RGN}\ \tau_r\ \tau_b}{\Delta; \Gamma; \mathcal{T} : \overline{\mathcal{T}} \vdash_{\text{exp}} \mathsf{bind}\ f : \tau_a \Leftarrow e_1; e_2 : \mathsf{RGN}\ \tau_r\ \tau_b}$$

The translation of an integer constant is a canonical example of allocation in the target calculus. The allocation is accomplished by the $\underline{\mathsf{newRGNVar}}$ command, applied to the appropriate region handle and value. However, the resulting command has type $\mathsf{RGN}\ \mathbb{T}_\tau^{\vdash_{\text{place}}}\left[\!\left[\rho\right]\!\right]\ (\mathsf{RGNVar}\ \mathbb{T}_\tau^{\vdash_{\text{place}}}\left[\!\left[\rho\right]\!\right]\ \mathsf{int})$, whereas the source typing

[3]Note that in the Single Effect Calculus, we only substitutes regions for region variables. This means that the sets of regions that appear in the program never change size (although they may change elements as a result of substitution). The $\mathbb{T}_\tau^{\vdash_{\text{place}}}\left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho_i\right]\!\right] \preceq \varrho$ translations require keeping the ordering of regions in a set $\{\rho_1, \ldots, \rho_n\}$ constant. It does not require a global ordering on region variables; such an ordering would not suffice for our purposes, because the ordering of elements in a set might change after substitution. Instead, we take $\{\rho_1, \ldots, \rho_n\}$ as a list with fixed order, where substitution preserves the order.

Translations yielding terms

Expressions

$$\mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\frac{\vdash_{\text{ctxt}} \Delta;\Gamma;\mathcal{S} : \overline{\mathcal{S}};\theta \quad \Delta;\overline{\mathcal{S}} \vdash_{\text{place}} \rho \qquad \Delta;\overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho}{\Delta;\Gamma;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} i \text{ at } \rho : (\mathsf{int},\rho),\theta} \right]\!\!\right] =$$

$$\mathbb{T}_v^{\vdash \text{rr}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho\right]\!\!\right] \; [\mathbb{T}_\tau^{\vdash \text{type}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{type}} (\mathsf{int},\rho)\right]\!\!\right]]$$
$$(\underline{\mathsf{newRGNVar}} \; [\mathbb{T}_\tau^{\vdash \text{place}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{place}} \rho\right]\!\!\right]]$$
$$[\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\!\right]]$$
$$\mathbb{T}_v^{\vdash \text{place}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{place}} \rho\right]\!\!\right]$$
$$i)$$

$$\mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\frac{\begin{array}{c}\Delta;\Gamma;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\mathsf{int},\rho_1),\theta \qquad \Delta;\overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho_1 \\ \Delta;\Gamma;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_2 : (\mathsf{int},\rho_2),\theta \qquad \Delta;\overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho_2 \\ \Delta;\overline{\mathcal{S}} \vdash_{\text{place}} \rho \qquad \Delta;\overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho\end{array}}{\Delta;\Gamma;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 \oplus e_2 \text{ at } \rho : (\mathsf{int},\rho),\theta} \right]\!\!\right] =$$

$$\mathsf{bind}\ a : \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{type}} (\mathsf{int},\rho_1)\right]\!\!\right] \Leftarrow \mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\Delta;\Gamma;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\mathsf{int},\rho_1),\theta\right]\!\!\right];$$
$$\mathsf{bind}\ a' : \mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\!\right] \Leftarrow \mathbb{T}_v^{\vdash \text{rr}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho_1\right]\!\!\right] \; [\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\!\right]]$$
$$(\underline{\mathsf{readRGNVar}} \; [\mathbb{T}_\tau^{\vdash \text{place}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{place}} \rho_1\right]\!\!\right]]$$
$$[\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\!\right]]$$
$$a);$$
$$\mathsf{bind}\ b : \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{type}} (\mathsf{int},\rho_2)\right]\!\!\right] \Leftarrow \mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\Delta;\Gamma;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_2 : (\mathsf{int},\rho_2),\theta\right]\!\!\right];$$
$$\mathsf{bind}\ b' : \mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\!\right] \Leftarrow \mathbb{T}_v^{\vdash \text{rr}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho_2\right]\!\!\right] \; [\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\!\right]]$$
$$(\underline{\mathsf{readRGNVar}} \; [\mathbb{T}_\tau^{\vdash \text{place}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{place}} \rho_2\right]\!\!\right]]$$
$$[\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\!\right]]$$
$$b);$$
$$\mathsf{let}\ z = a' \oplus b' \ \mathsf{in}$$
$$\mathbb{T}_v^{\vdash \text{rr}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho\right]\!\!\right] \; [\mathbb{T}_\tau^{\vdash \text{type}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{type}} (\mathsf{int},\rho)\right]\!\!\right]]$$
$$(\underline{\mathsf{newRGNVar}} \; [\mathbb{T}_\tau^{\vdash \text{place}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{place}} \rho\right]\!\!\right]]$$
$$[\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\!\right]]$$
$$\mathbb{T}_v^{\vdash \text{place}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{place}} \rho\right]\!\!\right]$$
$$z)$$

where a, a', b, b', z fresh

$$\mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\frac{\begin{array}{c}\Delta;\Gamma;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\mathsf{int},\rho_1),\theta \qquad \Delta;\overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho_1 \\ \Delta;\Gamma;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_2 : (\mathsf{int},\rho_2),\theta \qquad \Delta;\overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho_2\end{array}}{\Delta;\Gamma \vdash_{\text{exp}} e_1 \otimes e_2 : \mathsf{bool},\theta} \right]\!\!\right] =$$

$$\mathsf{bind}\ a : \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{type}} (\mathsf{int},\rho_1)\right]\!\!\right] \Leftarrow \mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\Delta;\Gamma;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\mathsf{int},\rho_1),\theta\right]\!\!\right];$$
$$\mathsf{bind}\ a' : \mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\!\right] \Leftarrow \mathbb{T}_v^{\vdash \text{rr}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho_1\right]\!\!\right] \; [\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\!\right]]$$
$$(\underline{\mathsf{readRGNVar}} \; [\mathbb{T}_\tau^{\vdash \text{place}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{place}} \rho_1\right]\!\!\right]]$$
$$[\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\!\right]]$$
$$a);$$
$$\mathsf{bind}\ b : \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{type}} (\mathsf{int},\rho_2)\right]\!\!\right] \Leftarrow \mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\Delta;\Gamma;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_2 : (\mathsf{int},\rho_2),\theta\right]\!\!\right];$$
$$\mathsf{bind}\ b' : \mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\!\right] \Leftarrow \mathbb{T}_v^{\vdash \text{rr}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho_2\right]\!\!\right] \; [\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\!\right]]$$
$$(\underline{\mathsf{readRGNVar}} \; [\mathbb{T}_\tau^{\vdash \text{place}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{place}} \rho_2\right]\!\!\right]]$$
$$[\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\!\right]]$$
$$b);$$
$$\mathsf{let}\ z = a' \otimes b' \ \mathsf{in}$$
$$\underline{\mathsf{returnRGN}} \; [\mathbb{T}_\tau^{\vdash \text{place}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{place}} \theta\right]\!\!\right]] \; [\mathbb{T}_\tau^{\vdash \text{type}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{type}} \mathsf{bool}\right]\!\!\right]] \; z$$

where a, a', b, b', z fresh

Figure 43: Translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$ (Terms (I))

Translations yielding terms

$$\mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\frac{\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \mathsf{tt} : \mathsf{bool}, \theta} \right]\!\!\right] = \underline{\mathsf{returnRGN}}\ [\mathbb{T}_\tau^{\vdash \text{place}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \theta]\!]]\ [\mathbb{T}_\tau^{\vdash \text{type}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \mathsf{bool}]\!]]\ \mathsf{tt}$$

$$\mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\frac{\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \mathsf{ff} : \mathsf{bool}, \theta} \right]\!\!\right] = \underline{\mathsf{returnRGN}}\ [\mathbb{T}_\tau^{\vdash \text{place}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \theta]\!]]\ [\mathbb{T}_\tau^{\vdash \text{type}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \mathsf{bool}]\!]]\ \mathsf{ff}$$

$$\mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\frac{\begin{array}{c}\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_b : \mathsf{bool}, \theta \\ \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_t : \tau, \theta \qquad \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_f : \tau, \theta\end{array}}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f : \tau, \theta} \right]\!\!\right] =$$

$$\begin{array}{l}\mathsf{bind}\ z : \mathbb{T}_\tau^{\vdash \text{type}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \mathsf{bool}]\!] \Leftarrow \mathbb{T}_e^{\vdash \text{exp}}\ [\![\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_b : \mathsf{bool}, \theta]\!]\ ; \\ \mathsf{if}\ z\ \mathsf{then}\ \mathbb{T}_e^{\vdash \text{exp}}\ [\![\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_t : \tau, \theta]\!]\ \mathsf{else}\ \mathbb{T}_e^{\vdash \text{exp}}\ [\![\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_f : \tau, \theta]\!]\end{array}$$

where z fresh

$$\mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\frac{\begin{array}{c}\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta \\ x \in dom(\Gamma) \qquad \Gamma(x) = \tau\end{array}}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} x : \tau, \theta} \right]\!\!\right] =$$

$$\underline{\mathsf{returnRGN}}\ [\mathbb{T}_\tau^{\vdash \text{place}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \theta]\!]]\ [\mathbb{T}_\tau^{\vdash \text{type}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau]\!]]\ x$$

$$\mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\frac{\begin{array}{c}\Delta; \Gamma, x : \tau_1; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e' : \tau_2, \theta' \\ \Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho\end{array}}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \lambda x : \tau_1.^{\theta'} e'\ \mathsf{at}\ \rho : (\tau_1 \xrightarrow{\theta'} \tau_2, \rho), \theta} \right]\!\!\right] =$$

$$\begin{array}{l}\mathbb{T}_v^{\vdash \text{rr}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho']\!]\ [\mathbb{T}_\tau^{\vdash \text{type}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} (\tau_1 \xrightarrow{\theta'} \tau_2, \rho')]\!]] \\ \quad (\underline{\mathsf{newRGNVar}}\ [\mathbb{T}_\tau^{\vdash \text{place}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho']\!]] \\ \qquad [\mathbb{T}_\tau^{\vdash \text{btype}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \tau_1 \xrightarrow{\theta'} \tau_2]\!]] \\ \qquad \mathbb{T}_v^{\vdash \text{place}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho']\!] \\ \qquad (\lambda x : \mathbb{T}_\tau^{\vdash \text{type}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_1]\!]\ .\mathbb{T}_e^{\vdash \text{exp}}\ [\![\Delta; \Gamma, x : \tau_1; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau_2, \theta']\!]))\end{array}$$

$$\mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\frac{\begin{array}{c}\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\tau_1 \xrightarrow{\theta'} \tau_2, \rho_1'), \theta \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho_1' \\ \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_2 : \tau_1, \theta \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \theta'\end{array}}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1\ e_2 : \tau_2, \theta} \right]\!\!\right] =$$

$$\begin{array}{l}\mathsf{bind}\ f : \mathbb{T}_\tau^{\vdash \text{type}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} (\tau_1 \xrightarrow{\theta'} \tau_2, \rho_1')]\!] \Leftarrow \mathbb{T}_e^{\vdash \text{exp}}\ [\![\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\tau_1 \xrightarrow{\theta'} \tau_2, \rho_1'), \theta]\!]\ ; \\ \mathsf{bind}\ g : \mathbb{T}_\tau^{\vdash \text{btype}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \tau_1 \xrightarrow{\theta'} \tau_2]\!] \Leftarrow \mathbb{T}_v^{\vdash \text{rr}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho_1']\!]\ [\mathbb{T}_\tau^{\vdash \text{btype}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \tau_1 \xrightarrow{\theta'} \tau_2]\!]] \\ \quad (\underline{\mathsf{readRGNVar}}\ [\mathbb{T}_\tau^{\vdash \text{place}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho_1']\!]] \\ \qquad [\mathbb{T}_\tau^{\vdash \text{btype}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \tau_1 \xrightarrow{\theta'} \tau_2]\!]] \\ \qquad f); \\ \mathsf{bind}\ a : \mathbb{T}_\tau^{\vdash \text{type}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_1]\!] \Leftarrow \mathbb{T}_e^{\vdash \text{exp}}\ [\![\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_2 : \tau_1, \theta]\!]\ ; \\ \mathsf{let}\ z = g\ a\ \mathsf{in} \\ \mathbb{T}_v^{\vdash \text{rr}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \theta']\!]\ [\mathbb{T}_\tau^{\vdash \text{type}}\ [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_2]\!]]\ z\end{array}$$

where f, g, a, z fresh

Figure 44: Translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$ (Terms (II))

Translations yielding terms

$$\mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\frac{\begin{array}{c}\Delta;\Gamma;\mathcal{S}:\overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : \tau_1, \theta \\ \Delta;\Gamma;\mathcal{S}:\overline{\mathcal{S}} \vdash_{\text{exp}} e_2 : \tau_2, \theta \\ \Delta;\overline{\mathcal{S}} \vdash_{\text{place}} \rho \qquad \Delta;\overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho\end{array}}{\Delta;\Gamma;\mathcal{S}:\overline{\mathcal{S}} \vdash_{\text{exp}} (e_1, e_2) \text{ at } \rho : (\tau_1 \times \tau_2, \rho), \theta} \right]\!\!\right] =$$

$$\begin{array}{l}\text{bind } a : \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{type}} \tau_1\right]\!\!\right] \Leftarrow \mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\Delta;\Gamma;\mathcal{S}:\overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : \tau_1, \theta\right]\!\!\right]; \\ \text{bind } b : \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{type}} \tau_2\right]\!\!\right] \Leftarrow \mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\Delta;\Gamma;\mathcal{S}:\overline{\mathcal{S}} \vdash_{\text{exp}} e_2 : \tau_2, \theta\right]\!\!\right]; \\ \mathbb{T}_v^{\vdash \text{rr}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho\right]\!\!\right] \ [\mathbb{T}_\tau^{\vdash \text{type}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{type}} (\tau_1 \times \tau_2, \rho)\right]\!\!\right]] \\ \qquad (\underline{\text{newRGNVar}} \ [\mathbb{T}_\tau^{\vdash \text{place}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{place}} \rho\right]\!\!\right]] \\ \qquad\quad [\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{btype}} \tau_1 \times \tau_2\right]\!\!\right]] \\ \qquad\quad \mathbb{T}_v^{\vdash \text{place}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{place}} \rho\right]\!\!\right] \\ \qquad\quad (a, b)) \\ \qquad\qquad \text{where } a, b \text{ fresh}\end{array}$$

$$\mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\frac{\begin{array}{c}\Delta;\Gamma;\mathcal{S}:\overline{\mathcal{S}} \vdash_{\text{exp}} e : (\tau_1 \times \tau_2, \rho), \theta \\ \Delta;\overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho\end{array}}{\Delta;\Gamma;\mathcal{S}:\overline{\mathcal{S}} \vdash_{\text{exp}} \text{fst } \ e : \tau_1, \theta} \right]\!\!\right] =$$

$$\begin{array}{l}\text{bind } x : \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{type}} (\tau_1 \times \tau_2, \rho)\right]\!\!\right] \Leftarrow \mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\Delta;\Gamma;\mathcal{S}:\overline{\mathcal{S}} \vdash_{\text{exp}} e : (\tau_1 \times \tau_2, \rho), \theta\right]\!\!\right]; \\ \text{bind } y : \mathbb{T}_\tau^{\vdash \text{bype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{bype}} \tau_1 \times \tau_2\right]\!\!\right] \Leftarrow \mathbb{T}_v^{\vdash \text{rr}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho\right]\!\!\right] \ [\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{btype}} \tau_1 \times \tau_2\right]\!\!\right]] \\ \qquad (\underline{\text{readRGNVar}} \ [\mathbb{T}_\tau^{\vdash \text{place}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{place}} \rho\right]\!\!\right]] \\ \qquad\quad [\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{btype}} \tau_1 \times \tau_2\right]\!\!\right]] \\ \qquad\quad a); \\ \text{let } z = \text{sel}_1 \ y \text{ in} \\ \underline{\text{returnRGN}} \ [\mathbb{T}_\tau^{\vdash \text{palce}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{place}} \theta\right]\!\!\right]] \ [\mathbb{T}_\tau^{\vdash \text{type}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{type}} \tau_1\right]\!\!\right]] \ z \\ \qquad\qquad \text{where } x, y, z \text{ fresh}\end{array}$$

$$\mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\frac{\begin{array}{c}\Delta;\Gamma;\mathcal{S}:\overline{\mathcal{S}} \vdash_{\text{exp}} e : (\tau_1 \times \tau_2, \rho), \theta \\ \Delta;\overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho\end{array}}{\Delta;\Gamma;\mathcal{S}:\overline{\mathcal{S}} \vdash_{\text{exp}} \text{snd } \ e : \tau_2, \theta} \right]\!\!\right] =$$

$$\begin{array}{l}\text{bind } x : \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{type}} (\tau_1 \times \tau_2, \rho)\right]\!\!\right] \Leftarrow \mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\Delta;\Gamma;\mathcal{S}:\overline{\mathcal{S}} \vdash_{\text{exp}} e : (\tau_1 \times \tau_2, \rho), \theta\right]\!\!\right]; \\ \text{bind } y : \mathbb{T}_\tau^{\vdash \text{bype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{bype}} \tau_1 \times \tau_2\right]\!\!\right] \Leftarrow \mathbb{T}_v^{\vdash \text{rr}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho\right]\!\!\right] \ [\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{btype}} \tau_1 \times \tau_2\right]\!\!\right]] \\ \qquad (\underline{\text{readRGNVar}} \ [\mathbb{T}_\tau^{\vdash \text{place}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{place}} \rho\right]\!\!\right]] \\ \qquad\quad [\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{btype}} \tau_1 \times \tau_2\right]\!\!\right]] \\ \qquad\quad a); \\ \text{let } z = \text{sel}_2 \ y \text{ in} \\ \underline{\text{returnRGN}} \ [\mathbb{T}_\tau^{\vdash \text{palce}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{place}} \theta\right]\!\!\right]] \ [\mathbb{T}_\tau^{\vdash \text{type}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{type}} \tau_2\right]\!\!\right]] \ z \\ \qquad\qquad \text{where } x, y, z \text{ fresh}\end{array}$$

$$\mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\frac{\begin{array}{c}\Delta;\overline{\mathcal{S}} \vdash_{\text{type}} \tau \qquad \vdash_{\text{ctxt}} \Delta;\Gamma;\mathcal{S}:\overline{\mathcal{S}};\theta \\ \Delta, \varrho \succeq \{\theta\};\Gamma;\mathcal{S}:\overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \varrho\end{array}}{\Delta;\Gamma;\mathcal{S}:\overline{\mathcal{S}} \vdash_{\text{exp}} \text{letregion } \varrho \text{ in } e : \tau, \theta} \right]\!\!\right] =$$

$$\begin{array}{l}\underline{\text{letRGN}} \ [\mathbb{T}_\tau^{\vdash \text{place}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{place}} \theta\right]\!\!\right]] \\ \qquad [\mathbb{T}_\tau^{\vdash \text{type}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{type}} \tau\right]\!\!\right]] \\ \qquad (\Lambda\varrho.\lambda w_\varrho : (\mathbb{T}_\tau^{\vdash \text{place}} \left[\!\!\left[\Delta;\overline{\mathcal{S}} \vdash_{\text{place}} \theta\right]\!\!\right] \preceq \varrho).\lambda h_\varrho : \text{RGNHandle } \varrho. \\ \qquad\ \mathbb{T}_e^{\vdash \text{exp}} \left[\!\!\left[\Delta, \varrho \succeq \{\theta\};\Gamma;\mathcal{S}:\overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \varrho\right]\!\!\right])\end{array}$$

Figure 45: Translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$ (Terms (III))

Translations yielding terms

$$\mathbb{T}_e^{\vdash_{\text{exp}}}\left[\!\!\left[\frac{\begin{array}{c}\Delta, \varrho \succeq \varphi; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} u' : \tau, \theta' \\ \Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho\end{array}}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \lambda \varrho \succeq \varphi.^{\theta'} u' \text{ at } \rho : (\Pi \varrho \succeq \varphi.^{\theta'} \tau, \rho), \theta}\right]\!\!\right] =$$

$$\mathbb{T}_v^{\vdash_{\text{rr}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho\right]\!\!\right]\ [\mathbb{T}_\tau^{\vdash_{\text{type}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} (\Pi \varrho \succeq \varphi.^{\theta'} \tau, \rho)\right]\!\!\right]]$$

$$(\underline{\mathsf{newRGNVar}}\ [\mathbb{T}_\tau^{\vdash_{\text{place}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho\right]\!\!\right]]$$

$$[\mathbb{T}_\tau^{\vdash_{\text{btype}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \Pi \varrho \succeq \varphi.^{\theta'} \tau\right]\!\!\right]]$$

$$\mathbb{T}_v^{\vdash_{\text{place}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho\right]\!\!\right]$$

$$(\Lambda \varrho.$$

$$\lambda w_\varrho : (\mathbb{T}_\tau^{\vdash_{\text{place}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho_1\right]\!\!\right] \preceq \varrho \times \cdots \times \mathbb{T}_\tau^{\vdash_{\text{place}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho_n\right]\!\!\right] \preceq \varrho).$$

$$\lambda h_\varrho : \mathsf{RGNHandle}\ \varrho.$$

$$\mathbb{T}_e^{\vdash_{\text{exp}}}\left[\!\!\left[\Delta, \varrho \succeq \varphi; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} u : \tau, \theta'\right]\!\!\right]))$$

where $\varphi = \{\rho_1, \ldots, \rho_n\}$

$$\mathbb{T}_e^{\vdash_{\text{exp}}}\left[\!\!\left[\frac{\begin{array}{c}\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\Pi \varrho \succeq \varphi.^{\theta'} \tau, \rho_1'), \theta \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \rho_1' \\ \Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho_2 \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{re}} \rho_2 \succeq \varphi \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \theta \succeq \theta'[\rho_2/\varrho]\end{array}}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1\ [\rho_2] : \tau[\rho_2/\varrho], \theta}\right]\!\!\right] =$$

$$\mathsf{bind}\ f : \mathbb{T}_\tau^{\vdash_{\text{type}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} (\Pi \varrho \succeq \varphi.^{\theta'} \tau, \rho_1')\right]\!\!\right] \Leftarrow \mathbb{T}_e^{\vdash_{\text{exp}}}\left[\!\!\left[\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : (\Pi \varrho \succeq \varphi.^{\theta'} \tau, \rho_1'), \theta\right]\!\!\right];$$

$$\mathsf{bind}\ g : \mathbb{T}_\tau^{\vdash_{\text{btype}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \Pi \varrho \succeq \varphi.^{\theta'} \tau\right]\!\!\right]$$

$$\Leftarrow \mathbb{T}_v^{\vdash_{\text{rr}}}\left[\!\!\left[\Delta; \mathcal{S} \vdash_{\text{rr}} \theta \succeq \rho_1'\right]\!\!\right]\ [\mathbb{T}_\tau^{\vdash_{\text{btype}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \Pi \varrho \succeq \varphi.^{\theta'} \tau\right]\!\!\right]]$$

$$(\underline{\mathsf{readRGNVar}}\ [\mathbb{T}_\tau^{\vdash_{\text{place}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho_1'\right]\!\!\right]]$$

$$[\mathbb{T}_\tau^{\vdash_{\text{btype}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \Pi \varrho \succeq \varphi.^{\theta'} \tau\right]\!\!\right]]$$

$$f);$$

$$\mathsf{let}\ z = (g\ [\mathbb{T}_\tau^{\vdash_{\text{place}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho_2\right]\!\!\right]]\ \mathbb{T}_v^{\vdash_{\text{re}}}\left[\!\!\left[\Delta; \mathcal{S} \vdash_{\text{re}} \rho_2 \succeq \varphi\right]\!\!\right]\ \mathbb{T}_v^{\vdash_{\text{place}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho_2\right]\!\!\right])\ \mathsf{in}$$

$$\mathbb{T}_v^{\vdash_{\text{rr}}}\left[\!\!\left[\Delta; \mathcal{S} \vdash_{\text{rr}} \theta \succeq \theta'\right]\!\!\right]\ \mathbb{T}_\tau^{\vdash_{\text{type}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau[\rho_2/\varrho]\right]\!\!\right]\ z$$

where f, g, z fresh

$$\mathbb{T}_e^{\vdash_{\text{exp}}}\left[\!\!\left[\frac{\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta \qquad r \in dom(\overline{\mathcal{S}}) \qquad l \in dom(\overline{\mathcal{S}}(r)) \qquad \mu = \mathcal{S}(r, l)}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \langle l \rangle_r : (\mu, r), \theta}\right]\!\!\right] =$$

$$\underline{\mathsf{returnRGN}}\ [\mathbb{T}_\tau^{\vdash_{\text{place}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \theta\right]\!\!\right]]\ [\mathbb{T}_\tau^{\vdash_{\text{type}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} (\mu, r)\right]\!\!\right]]\ \langle l \rangle_{s \sharp r}$$

$$\mathbb{T}_e^{\vdash_{\text{exp}}}\left[\!\!\left[\frac{\vdash_{\text{ctxt}} \Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}}; \theta \qquad \Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \mu}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \langle l \rangle_\bullet : (\mu, \bullet), \theta}\right]\!\!\right] =$$

$$\begin{cases} \underline{\mathsf{returnRGN}}\ [\mathbb{T}_\tau^{\vdash_{\text{place}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \theta\right]\!\!\right]]\ [\mathbb{T}_\tau^{\vdash_{\text{type}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} (\mu, \bullet)\right]\!\!\right]]\ \langle l \rangle_{\circ \sharp \bullet} & \text{if } \overline{\mathcal{S}} = \cdot \\ \underline{\mathsf{returnRGN}}\ [\mathbb{T}_\tau^{\vdash_{\text{place}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \theta\right]\!\!\right]]\ [\mathbb{T}_\tau^{\vdash_{\text{type}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} (\mu, \bullet)\right]\!\!\right]]\ \langle l \rangle_{s \sharp \bullet} & \text{otherwise} \end{cases}$$

Figure 46: Translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$ (Terms (IV))

Translations yielding storable values

Closed values

$$\mathbb{T}_v^{\vdash_{\text{cval}}}\left[\!\!\left[\frac{\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}}{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{cval}} \mathsf{tt} : \mathsf{bool}}\right]\!\!\right] = \mathsf{tt}$$

$$\mathbb{T}_v^{\vdash_{\text{cval}}}\left[\!\!\left[\frac{\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}}{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{cval}} \mathsf{ff} : \mathsf{bool}}\right]\!\!\right] = \mathsf{ff}$$

$$\mathbb{T}_v^{\vdash_{\text{cval}}}\left[\!\!\left[\frac{\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}} \qquad r \in dom(\overline{\mathcal{S}}) \qquad l \in dom(\overline{\mathcal{S}}(r)) \qquad \mu = \mathcal{S}(r,l)}{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{cval}} \langle l \rangle_r : (\mu, r)}\right]\!\!\right] = \langle l \rangle_{s\sharp r}$$

$$\mathbb{T}_v^{\vdash_{\text{cval}}}\left[\!\!\left[\frac{\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}} \qquad \cdot; \overline{\mathcal{S}} \vdash_{\text{btype}} \mu}{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{cval}} \langle l \rangle_\bullet : (\mu, \bullet)}\right]\!\!\right] = \begin{cases} \langle l \rangle_{\circ\sharp\bullet} & \text{if } \overline{\mathcal{S}} = \cdot \\ \langle l \rangle_{s\sharp\bullet} & \text{otherwise} \end{cases}$$

Figure 47: Translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$ (Closed values)

Translations yielding storable values

Storable values

$$\mathbb{T}_v^{\vdash_{\text{sto}}}\left[\!\!\left[\frac{\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}}{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{sto}} i : \mathsf{int}}\right]\!\!\right] = i$$

$$\mathbb{T}_v^{\vdash_{\text{sto}}}\left[\!\!\left[\frac{\cdot;\cdot, x : \tau_1; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e' : \tau_2, \theta'}{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{sto}} \lambda x : \tau_1.^{\theta'} e' : \tau_1 \xrightarrow{\theta'} \tau_2}\right]\!\!\right] = \lambda x : \mathbb{T}_\tau^{\vdash_{\text{type}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_1\right]\!\!\right].\mathbb{T}_e^{\vdash_{\text{exp}}}\left[\!\!\left[\cdot;\cdot, x : \tau_1; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau_2, \theta'\right]\!\!\right]$$

$$\mathbb{T}_v^{\vdash_{\text{sto}}}\left[\!\!\left[\frac{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{cval}} v_1 : \tau_1 \qquad \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{cval}} v_2 : \tau_2}{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{val}} (v_1, v_2) : \tau_1 \times \tau_2}\right]\!\!\right] = \left(\mathbb{T}_v^{\vdash_{\text{cval}}}\left[\!\!\left[\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{cval}} v_1 : \tau_1\right]\!\!\right], \mathbb{T}_v^{\vdash_{\text{cval}}}\left[\!\!\left[\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{val}} v_2 : \tau_2\right]\!\!\right]\right)$$

$$\mathbb{T}_v^{\vdash_{\text{sto}}}\left[\!\!\left[\frac{\cdot, \varrho \succeq \varphi; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} u' : \tau, \theta'}{\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{sto}} \lambda \varrho \succeq \varphi.^{\theta'} u' : \Pi \varrho \succeq \varphi.^{\theta'} \tau}\right]\!\!\right] =$$

$$\begin{array}{l} \Lambda\varrho. \\ \lambda w_\varrho : (\mathbb{T}_\tau^{\vdash_{\text{place}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho_1\right]\!\!\right] \preceq \varrho \times \cdots \times \mathbb{T}_\tau^{\vdash_{\text{place}}}\left[\!\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho_n\right]\!\!\right] \preceq \varrho). \\ \lambda h_\varrho : \mathsf{RGNHandle}\ \varrho. \\ \mathbb{T}_e^{\vdash_{\text{exp}}}\left[\!\!\left[\cdot, \varrho \succeq \varphi; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} u : \tau, \theta'\right]\!\!\right] \end{array}$$

Figure 48: Translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$ (Storable values)

Translations yielding stacks

Stacks

$$\mathbb{T}_S^{\vdash_{\text{stack}}} [\![\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}]\!] \quad = \quad S^*$$

where

$$dom(S) = dom(S^*)$$

$$\forall r \in dom(S).dom(S(r)) = dom(S^*(r))$$

$$\forall r \in dom(S).\forall l \in dom(\overline{\mathcal{S}}(r)).S^*(r,l) = \mathbb{T}_v^{\vdash_{\text{sto}}} [\![\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{sto}} S(r,l) : \mathcal{S}(r,l)]\!]$$

Translations yielding towers

Stacks

$$\mathbb{T}_T^{\vdash_{\text{stack}}} [\![\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}]\!] \quad = \quad \begin{cases} \cdot & \text{if } S = \cdot \\ \cdot, s \mapsto \mathbb{T}_S^{\vdash_{\text{stack}}} [\![\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}]\!] & \text{otherwise} \end{cases}$$

Figure 49: Translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$ (Stacks (III))

Translations yielding terms

Programs

$$\mathbb{T}_e^{\vdash_{\text{prog}}} \left[\!\!\left[\frac{\cdot, \mathcal{H} \succeq \{\}; \cdot; \cdot : \cdot \vdash_{\text{exp}} p : \mathsf{bool}, \mathcal{H}}{\vdash_{\text{prog}} p \ \mathsf{ok}} \right]\!\!\right] =$$

$$\mathsf{runRGN}\ [\mathbb{T}_\tau^{\vdash_{\text{type}}} [\![\cdot, \mathcal{H} \succeq \{\}; \cdot; \cdot : \cdot \vdash_{\text{type}} \mathsf{bool}]\!]]\ (\Lambda\mathcal{H}.\lambda h_{\mathcal{H}} : \mathsf{RGNHandle}\ \mathcal{H}.$$

$$\mathsf{let}\ w_{\mathcal{H}} = ()\ \mathsf{in}$$

$$\mathbb{T}_e^{\vdash_{\text{exp}}} [\![\cdot, \mathcal{H} \succeq \{\}; \cdot; \cdot : \cdot \vdash_{\text{exp}} p : \mathsf{bool}, \mathcal{H}]\!])$$

Figure 50: Translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$ (Programs)

judgement requires the computation to be expressed relative to the region θ. We coerce the computation using a witness function, whose existence is implied by the judgement $\Delta; \overline{\mathsf{S}} \vdash_{\mathrm{rr}} \rho \succeq \rho'$. Allocation of a function proceeds in exactly the same manner. Function application, while notationally heavy, is simple. The thenRGN commands (implicit in the bind expressions) sequence evaluating the function to a location, reading the location, evaluating the argument, and applying the function to the argument.

The translation of letregion $\varrho.e$ is pleasantly direct. As described above, we introduce ϱ, ϱ^h, and ϱ^w through Λ- and λ-abstractions. The region handle and coercion function are supplied by the letRGN command when the computation is executed.

The translation of region abstraction is similar to the translation of functions. Once again, region handles and witness functions are λ-bound in accordance to the invariants described above. During the translation of region applications, the appropriate tuple of witness functions (constructed by $\mathbb{T}^{\vdash_{\mathrm{re}}}_{e}[\![\cdot]\!]$) and region handle are supplied as arguments.

Figures 47 and 48 give the translations of closed and storable values, which follow directly from the translations of expressions. Figure 49 gives the translation of stacks, where each stored value is translated according to the $\vdash_{\mathrm{sto}}$ derivation implied by the $\vdash_{\mathrm{stack}}$ derivation.

Figure 50 shows the translation of programs. An entire region computation is encapsulated and run by the runRGN expression. We bind $\mathcal{H}^w$ to an empty tuple, which corresponds to the absence of any coercion functions to the region $\mathcal{H}$.

4.1 Surface programs

Once again, things are complicated by the distinction between surface and computation syntax. All of the translations given in this section must be given via translations on derivations, essentially to propagate the $\overline{\mathsf{S}}$ stack domain to each point where a $\bullet$ may appear. The difficulty is that during the evaluation of the translation of a source program, there can be points at which $\bullet$ is translated to either $s\sharp\bullet$ or $\circ\sharp\bullet$. We make this choice based on whether or not *any* region is in the $\overline{\mathsf{S}}$ stack domain.

When we are only interested in translating surface programs, then neither r nor $\bullet$ can appear in types or expressions. This simplifies many of the translations. We no longer require any translations of stack domains, stack types, or stacks. Furthermore, the translations of region contexts, places (which must be of the form ϱ), types and boxed types, and value contexts can all be given as syntactic translations (rather than translations on derivations).

4.2 Type Preservation

Lemma 39 (Translation preserves types)

(1) *If* $\vdash_{\text{rom}} \overline{\mathcal{R}}$,
then $\vdash_{\text{rdom}} \mathbb{T}^{\vdash_{\text{rdom}}}_{\overline{\mathcal{R}}} \left[\!\left[\vdash_{\text{rdom}} \overline{\mathcal{R}}\right]\!\right]$.
Furthermore, $dom(\overline{\mathcal{R}}) = dom(\mathbb{T}^{\vdash_{\text{rdom}}}_{\overline{\mathcal{R}}} \left[\!\left[\vdash_{\text{rdom}} \overline{\mathcal{R}}\right]\!\right])$.

(2) *If* $\vdash_{\text{sdom}} \overline{\mathcal{S}}$,
then $\vdash_{\text{sdom}} \mathbb{T}^{\vdash_{\text{sdom}}}_{\overline{\mathcal{S}}} \left[\!\left[\vdash_{\text{sdom}} \overline{\mathcal{S}}\right]\!\right]$.
Furthermore, $dom(\overline{\mathcal{S}}) = dom(\mathbb{T}^{\vdash_{\text{sdom}}}_{\overline{\mathcal{S}}} \left[\!\left[\vdash_{\text{sdom}} \overline{\mathcal{S}}\right]\!\right])$.

(3) *If* $\vdash_{\text{sdom}} \overline{\mathcal{S}}$,
then $\vdash_{\text{tdom}} \mathbb{T}^{\vdash_{\text{sdom}}}_{\overline{\mathcal{T}}} \left[\!\left[\vdash_{\text{rdom}} \overline{\mathcal{S}}\right]\!\right]$.

(4) *If* $\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta$,
then $\vdash_{\text{tctxt}} \mathbb{T}^{\vdash_{\text{rctxt}}}_{\Delta} \left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right]$.
Furthermore, $dom(\Delta) = dom(\mathbb{T}^{\vdash_{\text{rctxt}}}_{\Delta} \left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right])$.

(5) *If* $\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho$,
then $\mathbb{T}^{\vdash_{\text{rctxt}}}_{\Delta} \left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right]; \mathbb{T}^{\vdash_{\text{sdom}}}_{\overline{\mathcal{T}}} \left[\!\left[\vdash_{\text{sdom}} \overline{\mathcal{S}}\right]\!\right] \vdash_{\text{type}} \mathbb{T}^{\vdash_{\text{place}}}_{\tau} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho\right]\!\right]$.

(6) *If* $\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta$,
then $\mathbb{T}^{\vdash_{\text{rctxt}}}_{\Delta} \left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right]; \mathbb{T}^{\vdash_{\text{sdom}}}_{\overline{\mathcal{T}}} \left[\!\left[\vdash_{\text{rdom}} \overline{\mathcal{S}}\right]\!\right] \vdash_{\text{vctxt}} \mathbb{T}^{\vdash_{\text{rctxt}}}_{\Gamma} \left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right]$.

(7) *If* $\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \mu$,
then $\mathbb{T}^{\vdash_{\text{rctxt}}}_{\Delta} \left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right]; \mathbb{T}^{\vdash_{\text{sdom}}}_{\overline{\mathcal{T}}} \left[\!\left[\vdash_{\text{sdom}} \overline{\mathcal{S}}\right]\!\right] \vdash_{\text{type}} \mathbb{T}^{\vdash_{\text{btype}}}_{\tau} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{btype}} \mu\right]\!\right]$.

(8) *If* $\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau$,
then $\mathbb{T}^{\vdash_{\text{rctxt}}}_{\Delta} \left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right]; \mathbb{T}^{\vdash_{\text{sdom}}}_{\overline{\mathcal{T}}} \left[\!\left[\vdash_{\text{sdom}} \overline{\mathcal{S}}\right]\!\right] \vdash_{\text{type}} \mathbb{T}^{\vdash_{\text{type}}}_{\tau} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau\right]\!\right]$.

(9) *If* $\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}$,
then $\vdash_{\text{ttype}} \mathbb{T}^{\vdash_{\text{stype}}}_{\overline{\mathcal{T}}} \left[\!\left[\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}\right]\!\right]$.

(10) *If* $\Delta; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \Gamma$,
then $\mathbb{T}^{\vdash_{\text{rctxt}}}_{\Delta} \left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right]; \mathbb{T}^{\vdash_{\text{sdom}}}_{\overline{\mathcal{T}}} \left[\!\left[\vdash_{\text{rdom}} \overline{\mathcal{S}}\right]\!\right] \vdash_{\text{vctxt}} \mathbb{T}^{\vdash_{\text{vctxt}}}_{\Gamma} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \Gamma\right]\!\right]$.

(11) *If* $\Delta; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \Gamma$,
then $\mathbb{T}^{\vdash_{\text{rctxt}}}_{\Delta} \left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right]; \mathbb{T}^{\vdash_{\text{sdom}}}_{\overline{\mathcal{T}}} \left[\!\left[\vdash_{\text{rdom}} \overline{\mathcal{S}}\right]\!\right] \vdash_{\text{vctxt}} \mathbb{T}^{\vdash_{\text{rctxt}}}_{\Gamma} \left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right], \mathbb{T}^{\vdash_{\text{vctxt}}}_{\Gamma} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \Gamma\right]\!\right]$.

(12) *If* $\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho'$,
then $\mathbb{T}^{\vdash_{\text{rctxt}}}_{\Delta} \left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right]; \mathbb{T}^{\vdash_{\text{sdom}}}_{\overline{\mathcal{T}}} \left[\!\left[\vdash_{\text{sdom}} \overline{\mathcal{S}}\right]\!\right] \vdash_{\text{type}} \mathbb{T}^{\vdash_{\text{rr}}}_{\tau} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho'\right]\!\right]$.

(13) *If* $\Delta; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \varphi$,
then $\mathbb{T}^{\vdash_{\text{rctxt}}}_{\Delta} \left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right]; \mathbb{T}^{\vdash_{\text{sdom}}\overline{\mathcal{S}}}_{\overline{\mathcal{T}}} \left[\!\left[\vdash_{\text{sdom}} \overline{\mathcal{S}}\right]\!\right] \vdash_{\text{type}} \mathbb{T}^{\vdash_{\text{re}}}_{\tau} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \varphi\right]\!\right]$.

(14) *If* $\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}$ *and* $\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho'$,
then $\mathbb{T}^{\vdash_{\text{rctxt}}}_{\Delta} \left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right]; \mathbb{T}^{\vdash_{\text{rctxt}}}_{\Gamma} \left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right]; \mathbb{T}^{\vdash_{\text{stype}}}_{\mathcal{T}} \left[\!\left[\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}\right]\!\right] : \mathbb{T}^{\vdash_{\text{sdom}}}_{\overline{\mathcal{T}}} \left[\!\left[\vdash_{\text{sdom}} \overline{\mathcal{S}}\right]\!\right] \vdash_{\text{exp}} \mathbb{T}^{\vdash_{\text{rr}}}_{v} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho'\right]\!\right] : \mathbb{T}^{\vdash_{\text{rr}}}_{\tau} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{rr}} \rho \succeq \rho'\right]\!\right]$.

(15) *If* $\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}$ *and* $\Delta; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \varphi$,
then $\mathbb{T}_{\Delta}^{\vdash_{\text{rctxt}}} \left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right]; \mathbb{T}_{\Gamma}^{\vdash_{\text{rctxt}}} \left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right]; \mathbb{T}_{\mathcal{T}}^{\vdash_{\text{stype}}} \left[\!\left[\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}\right]\!\right] : \mathbb{T}_{\overline{\mathcal{T}}}^{\vdash_{\text{sdom}}} \left[\!\left[\vdash_{\text{sdom}} \overline{\mathcal{S}}\right]\!\right] \vdash_{\text{type}} \mathbb{T}_{v}^{\vdash_{\text{re}}} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \varphi\right]\!\right] : \mathbb{T}_{\tau}^{\vdash_{\text{re}}} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \varphi\right]\!\right]$.

(16) *If* $\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}$ *and* $\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho$,
then $\mathbb{T}_{\Delta}^{\vdash_{\text{rctxt}}} \left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right]; \mathbb{T}_{\Gamma}^{\vdash_{\text{rctxt}}} \left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right]; \mathbb{T}_{\mathcal{T}}^{\vdash_{\text{stype}}} \left[\!\left[\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}\right]\!\right] : \mathbb{T}_{\overline{\mathcal{T}}}^{\vdash_{\text{sdom}}} \left[\!\left[\vdash_{\text{sdom}} \overline{\mathcal{S}}\right]\!\right] \vdash_{\text{exp}} \mathbb{T}_{v}^{\vdash_{\text{place}}} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho\right]\!\right] : \mathsf{handle}(\mathbb{T}_{\tau}^{\vdash_{\text{place}}} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{place}} \rho\right]\!\right])$.

(17) *If* $\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta$,
then $\mathbb{T}_{\Delta}^{\vdash_{\text{rctxt}}} \left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right]; \mathbb{T}_{\Gamma}^{\vdash_{\text{rctxt}}} \left[\!\left[\overline{\mathcal{S}} \vdash_{\text{rctxt}} \Delta\right]\!\right], \mathbb{T}_{\Gamma}^{\vdash_{\text{vctxt}}} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{vctxt}} \Gamma\right]\!\right]; \mathbb{T}_{\mathcal{T}}^{\vdash_{\text{stype}}} \left[\!\left[\vdash_{\text{sdom}} \mathcal{S} : \overline{\mathcal{S}}\right]\!\right] : \mathbb{T}_{\overline{\mathcal{T}}}^{\vdash_{\text{sdom}}} \left[\!\left[\vdash_{\text{sdom}} \overline{\mathcal{S}}\right]\!\right] \vdash_{\text{exp}} \mathbb{T}_{e}^{\vdash_{\text{exp}}} \left[\!\left[\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta\right]\!\right] : \mathsf{RGN}\ \mathbb{T}_{\tau}^{\vdash_{\text{place}}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} \theta\right]\!\right]\ \mathbb{T}_{\tau}^{\vdash_{\text{type}}} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\text{type}} \tau\right]\!\right]$.

(18) *If* $\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{cval}} v : \tau$,
then $\cdot; \cdot; \mathbb{T}_{\mathcal{T}}^{\vdash_{\text{stype}}} \left[\!\left[\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}\right]\!\right] : \mathbb{T}_{\overline{\mathcal{T}}}^{\vdash_{\text{sdom}}} \left[\!\left[\vdash_{\text{sdom}} \overline{\mathcal{S}}\right]\!\right] \vdash_{\text{exp}} \mathbb{T}_{v}^{\vdash_{\text{cval}}} \left[\!\left[\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{cval}} v : \tau\right]\!\right] : \mathbb{T}_{\tau}^{\vdash_{\text{type}}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau\right]\!\right]$.

(19) *If* $\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{sto}} w : \tau$,
then $\cdot; \cdot; \mathbb{T}_{\mathcal{T}}^{\vdash_{\text{stype}}} \left[\!\left[\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}\right]\!\right] : \mathbb{T}_{\overline{\mathcal{T}}}^{\vdash_{\text{sdom}}} \left[\!\left[\vdash_{\text{sdom}} \overline{\mathcal{S}}\right]\!\right] \vdash_{\text{exp}} \mathbb{T}_{v}^{\vdash_{\text{sto}}} \left[\!\left[\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{sto}} w : \mu\right]\!\right] : \mathbb{T}_{\tau}^{\vdash_{\text{btype}}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{btype}} \mu\right]\!\right]$.

(20) *If* $\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}$,
then $\vdash_{\text{tower}} \mathbb{T}_{T}^{\vdash_{\text{stack}}} \left[\!\left[\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}\right]\!\right] : \mathbb{T}_{\mathcal{T}}^{\vdash_{\text{stype}}} \left[\!\left[\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}\right]\!\right] : \mathbb{T}_{\mathcal{T}}^{\vdash_{\text{sdom}}} \left[\!\left[\vdash_{\text{sdom}} \overline{\mathcal{S}}\right]\!\right]$.

(21) *If* $\vdash_{\text{prog}} p\ \mathsf{ok}$,
then $\cdot; \cdot; \cdot; \cdot \vdash_{\text{exp}} \mathbb{T}_{e}^{\vdash_{\text{prog}}} \left[\!\left[\cdot, \mathcal{H} \succeq \{\}; \cdot; \cdot : \cdot \vdash_{\text{exp}} p : \mathsf{bool}, \mathcal{H}\right]\!\right] : \mathbb{T}_{\tau}^{\vdash_{\text{type}}} \left[\!\left[\cdot, \mathcal{H} \succeq \{\}; \cdot \vdash_{\text{type}} \mathsf{bool}\right]\!\right]$.

Proof. By (mutual) induction on the derivations, making frequent appeals to the well-formedness lemmas of Section 2.7.1. □

4.3 Semantics Preservation

In this section, we prove that the translation is meaning preserving. The essence of the proof relies on a *coherence* lemma stating that the translation of witnesses yields functions that are operationally equivalent to the identity function.

Lemma 40 (Coherence)

Suppose $\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}$ *and* $\cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r \succeq r_i$.
Let $\mathbb{T}_{\overline{\mathcal{F}}}^{\vdash_{\text{sdom}}} \left[\!\left[\vdash_{\text{sdom}} \overline{\mathcal{S}}\right]\!\right] = \cdot, s \mapsto \overline{\mathcal{S}}^*$, $\mathbb{T}_{\mathcal{T}}^{\vdash_{\text{stype}}} \left[\!\left[\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}\right]\!\right] = \cdot, s \mapsto \mathcal{S}^*$, $\mathbb{T}_{T}^{\vdash_{\text{stack}}} \left[\!\left[\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}\right]\!\right] = \cdot, s \mapsto S^*$, *and* $\mathbb{T}_{v}^{\vdash_{\text{rr}}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r \succeq r_i\right]\!\right] = v_w^*$.
If $\cdot; \cdot; \cdot, s \mapsto \mathcal{S}^* : \cdot, s \mapsto \overline{\mathcal{S}}^* \vdash_{\text{exp}} \kappa^{v*} : \mathsf{RGN}\ s\sharp r_i\ \tau_a$ *and* $\cdot, s \mapsto S^*; \kappa^{v*} \hookrightarrow_\kappa S'^*; v'^*$,
then $\cdot, s \mapsto S^*; v_w^*\ [\tau_a]\ \kappa^{v*} \hookrightarrow \kappa^{v*\prime}$ *and* $\cdot, s \mapsto S^*; \kappa^{v*\prime} \hookrightarrow_\kappa S'^*; v'^*$.

Proof. By applying Lemma 7 to $\cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r \succeq r_i$, we conclude that $\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} r$ and $\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} r_i$. Hence, we conclude that $r \in dom(\overline{\mathcal{S}})$ and $r_i \in dom(\overline{\mathcal{S}})$. Furthermore, $r \in dom(\mathcal{S})$ and $r_i \in dom(\mathcal{S})$ and $r \in dom(S)$ and $r_i \in dom(S)$. Thus, $\overline{\mathcal{S}} \neq \cdot$, $\mathcal{S} \neq \cdot$, and $S \neq \cdot$. Hence, $\mathbb{T}_{\overline{\mathcal{F}}}^{\vdash_{\text{sdom}}} \left[\!\left[\vdash_{\text{sdom}} \overline{\mathcal{S}}\right]\!\right] = \cdot, s \mapsto \mathbb{T}_{\overline{\mathcal{S}}}^{\vdash_{\text{sdom}}} \left[\!\left[\vdash_{\text{sdom}} \overline{\mathcal{S}}\right]\!\right] = \cdot, s \mapsto \overline{\mathcal{S}}^*$, $\mathbb{T}_{\mathcal{T}}^{\vdash_{\text{stype}}} \left[\!\left[\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}\right]\!\right] = \cdot, s \mapsto \mathbb{T}_{\mathcal{S}}^{\vdash_{\text{stype}}} \left[\!\left[\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}\right]\!\right] = \cdot, s \mapsto \mathcal{S}^*$, and $\mathbb{T}_{T}^{\vdash_{\text{stack}}} \left[\!\left[\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}\right]\!\right] = \cdot, s \mapsto \mathbb{T}_{S}^{\vdash_{\text{stack}}} \left[\!\left[\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}\right]\!\right] = \cdot, s \mapsto S^*$ as assumed.

By applying Lemma 39 to $\cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r \succeq r_i$, we conclude that $\cdot; \cdot; \cdot, s \mapsto \mathcal{S}^* : \cdot, s \mapsto \overline{\mathcal{S}}^* \vdash_{\text{exp}} v_w^* : s\sharp r_i \preceq s\sharp r$.
Note that

$$\dfrac{\dfrac{\boxed{\cdot; \cdot, s \mapsto \overline{\mathcal{S}}^* \vdash_{\text{type}} \tau_a} \qquad \boxed{\cdot; \cdot; \cdot, s \mapsto \mathcal{S}^* : \cdot, s \mapsto \overline{\mathcal{S}}^* \vdash_{\text{exp}} v_w^* : \forall\beta.\mathsf{RGN}\ s\sharp r_i\ \beta \longrightarrow \mathsf{RGN}\ s\sharp r\ \beta}}{\cdot; \cdot; \cdot, s \mapsto \mathcal{S}^* : \cdot, s \mapsto \overline{\mathcal{S}}^* \vdash_{\text{exp}} v_w^*\ [\tau_a] : \mathsf{RGN}\ s\sharp r_i\ \tau_a \longrightarrow \mathsf{RGN}\ s\sharp r\ \tau_a} \qquad \boxed{\cdot; \cdot; \cdot, s \mapsto \mathcal{S}^* : \cdot, s \mapsto \overline{\mathcal{S}}^* \vdash_{\text{exp}} \kappa^{v*} : \mathsf{RGN}\ s\sharp r_i\ \tau_a}}{\cdot; \cdot; \cdot, s \mapsto \mathcal{S}^* : \cdot, s \mapsto \overline{\mathcal{S}}^* \vdash_{\text{exp}} v_w^*\ [\tau_a]\ \kappa^{v*} : \mathsf{RGN}\ s\sharp r_i\ \tau_a}$$

Proceed by induction on the derivation of $\cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r \succeq r_i$.

Case $\dfrac{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \cdot}{\cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r \succeq r_i}$: By inspection of the derivation $\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}$, we conclude that $\mathcal{S} \equiv \mathcal{S}_1, r_i \mapsto \mathcal{R}_i, \mathcal{S}_2, r_i \mapsto \mathcal{R}, \mathcal{S}_3$ and $S \equiv S_1, r_i \mapsto R_i, S_2, r_i \mapsto R, S_3$.
By applying Lemma 39, we conclude that $\overline{\mathcal{S}}^* \equiv \overline{\mathcal{S}_1}^*, r_i \mapsto \overline{\mathcal{R}_i}^*, \overline{\mathcal{S}_2}^*, r_i \mapsto \overline{\mathcal{R}}^*, \overline{\mathcal{S}_3}^*$, $\mathcal{S}^* \equiv \mathcal{S}_1{}^*, r_i \mapsto \mathcal{R}_i{}^*, \mathcal{S}_2{}^*, r \mapsto \mathcal{R}^*, \mathcal{S}_3{}^*$, and $S^* \equiv S_1{}^*, r_i \mapsto R_i{}^*, S_2{}^*, r \mapsto R^*, S_3{}^*$.
Note that

$$v_w^* \quad = \quad \Lambda\beta.\lambda k : \mathsf{RGN}\ s\sharp r_i\ \beta.\underline{\mathsf{witnessRGN}}\ s\sharp r_i\ s\sharp r\ [\beta]\ k$$

Note that

$$\dfrac{\boxed{\cdot, s \mapsto S^*; v_w^*\ [\tau_a] \hookrightarrow \lambda k : \mathsf{RGN}\ s\sharp r_i\ \tau_a.\underline{\mathsf{witnessRGN}}\ s\sharp r_i\ s\sharp r\ [\tau_a]\ k} \qquad \boxed{\cdot, s \mapsto S^*; \kappa^{v*} \hookrightarrow \kappa^{v*}} \qquad \boxed{\cdot, s \mapsto S^*; (\underline{\mathsf{witnessRGN}}\ s\sharp r_i\ s\sharp r\ [\tau_a]\ k)[\kappa^{v*}/k] \hookrightarrow \underline{\mathsf{witnessRGN}}\ s\sharp r_i\ s\sharp r\ [\tau_a]\ \kappa^{v*}}}{\cdot, s \mapsto S^*; v_w^*\ [\tau_a]\ \kappa^{v*} \hookrightarrow \underline{\mathsf{witnessRGN}}\ s\sharp r_i\ s\sharp r\ [\tau_a]\ \kappa^{v*}}$$

and

$$\dfrac{\boxed{S^* \equiv S_1{}^*, r_i \mapsto R_i{}^*, S_2{}^*, r_i \mapsto R^*, S_3{}^*} \qquad \boxed{\cdot, s \mapsto S^*; \kappa^{v*} \hookrightarrow_\kappa S'^*; v'^*}}{\cdot, s \mapsto S^*; \underline{\mathsf{witnessRGN}}\ s\sharp r_i\ s\sharp r\ [\tau_a]\ \kappa^{v*} \hookrightarrow_\kappa S'^*; v'^*}$$

Case $\dfrac{\overline{\mathcal{S}} \vdash_{\text{rctxt}} \cdot \qquad \cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r \succeq r_i \quad {\scriptstyle i \in 1 \ldots n}}{\cdot; \overline{\mathcal{S}} \vdash_{\text{re}} r \succeq \{r_1, \ldots, r_n\}}{\scriptstyle \cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r \succeq r_i}$: Applying the induction hypothesis to $\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}$ and $\cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r \succeq r_i$, we conclude that $\cdot, s \mapsto S^*; \mathbb{T}_{v}^{\vdash_{\text{rr}}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq r_i\right]\!\right]\ [\tau_a]\ \kappa^{v*} \hookrightarrow \kappa^{v*\prime}$ and $\cdot, s \mapsto S^*; \kappa^{v*\prime} \hookrightarrow_\kappa S'^*; v'^*$.

Note that

$$v_w^* \quad = \quad \Lambda\beta.\lambda k : \mathsf{RGN}\ \mathbb{T}_\tau^{\vdash \mathrm{place}} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{place}} r_i\right]\!\right]\ \beta.\mathsf{let}\ w = \mathsf{sel}_i\ \mathbb{T}_v^{\vdash \mathrm{re}} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{re}} r \succeq \{r_1, \ldots, r_n\}\right]\!\right]\ \mathsf{in}\ w\ [\beta]\ k$$

Note that

$$\dfrac{\dfrac{\boxed{\cdot, s \mapsto S^*; v_w \hookrightarrow \Lambda\beta.\lambda k : \mathsf{RGN}\ \mathbb{T}_\tau^{\vdash \mathrm{place}} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{place}} r_i\right]\!\right]\ \beta.\mathsf{let}\ w = \mathsf{sel}_i\ \mathbb{T}_v^{\vdash \mathrm{re}} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{re}} r \succeq \{r_1, \ldots, r_n\}\right]\!\right]\ \mathsf{in}\ w\ [\beta]\ k}}{\cdot, s \mapsto S^*; v_w\ [\tau_a] \hookrightarrow \lambda k : \mathsf{RGN}\ \mathbb{T}_\tau^{\vdash \mathrm{place}} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{place}} r_i\right]\!\right]\ \tau_a.\mathsf{let}\ w = \mathsf{sel}_i\ \mathbb{T}_v^{\vdash \mathrm{re}} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{re}} r \succeq \{r_1, \ldots, r_n\}\right]\!\right]\ \mathsf{in}\ w\ [\tau_a]\ k}\quad \boxed{\cdot, s \mapsto S^*; {\kappa_v}^* \hookrightarrow {\kappa_v}^*}}{\cdot, s \mapsto S^*; v_w\ [\tau_a]\ {\kappa_v}^* \hookrightarrow {\kappa^v}^{*\prime}}$$

where the second premise is

$$\dfrac{\dfrac{\boxed{\cdot, s \mapsto S^*; \mathbb{T}_v^{\vdash \mathrm{re}} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{re}} r \succeq \{r_1, \ldots, r_n\}\right]\!\right] \hookrightarrow (\mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r \succeq r_1\right]\!\right], \ldots, \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r \succeq r_n\right]\!\right])}}{\cdot, s \mapsto S^*; \mathsf{sel}_i\ \mathbb{T}_v^{\vdash \mathrm{re}} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{re}} r \succeq \{r_1, \ldots, r_n\}\right]\!\right] \hookrightarrow \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r \succeq r_i\right]\!\right]}\quad \boxed{\cdot, s \mapsto S^*; \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r' \succeq r_i\right]\!\right]\ [\tau_a]\ {\kappa^v}^* \hookrightarrow {\kappa^v}^{*\prime}}}{\cdot, s \mapsto S^*; \mathsf{let}\ w = \mathsf{sel}_i\ \mathbb{T}_v^{\vdash \mathrm{re}} \left[\!\left[\Delta; \overline{\mathcal{S}} \vdash_{\mathrm{re}} r \succeq \{r_1, \ldots, r_n\}\right]\!\right]\ \mathsf{in}\ w\ [\tau_a]\ {\kappa_v}^* \hookrightarrow {\kappa^v}^{*\prime}}$$

and $\cdot, s \mapsto S^*; {\kappa^v}^{*\prime} \hookrightarrow_\kappa S'^*; v'^*$.

Case $\dfrac{\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{place}} r_i}{\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r_i \succeq r_i}$: Note that

$$v_w^* \quad = \quad \Lambda\beta.\lambda k : \mathsf{RGN}\ s\sharp r_i\ \beta.k$$

Note that

$$\dfrac{\boxed{\cdot, s \mapsto S^*; v_w^*\ [\tau_a] \hookrightarrow \lambda k : \mathsf{RGN}\ s\sharp r_i\ \tau_a.k} \quad \boxed{\cdot, s \mapsto S^*; {\kappa^v}^* \hookrightarrow {\kappa^v}^*} \quad \boxed{\cdot, s \mapsto S^*; k[{\kappa^v}^*/k] \hookrightarrow {\kappa^v}^*}}{\cdot, s \mapsto S^*; v_w^*\ [\tau_a]\ {\kappa^v}^* \hookrightarrow {\kappa^v}^*}$$

and

$$\dfrac{}{\cdot, s \mapsto S^*; {\kappa^v}^* \hookrightarrow_\kappa S'^*; v'^*}$$

Case $\dfrac{\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r \succeq \rho' \qquad \cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} \rho' \succeq r_i}{\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r \succeq r_i}$: By applying Lemma 17 to $\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r \succeq \rho'$, we conclude that $\rho' \equiv r'$.

Hence,

$$\dfrac{\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r \succeq r' \qquad \cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r' \succeq r_i}{\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r \succeq r_i}$$

Note that

$$v_w^* \quad = \quad \Lambda\beta.\lambda k : \mathsf{RGN}\ s\sharp r_i\ \beta.\mathsf{let}\ k' = \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r' \succeq r_i\right]\!\right]\ [\beta]\ k\ \mathsf{in}\ \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r \succeq r'\right]\!\right]\ [\beta]\ k'$$

Applying the induction hypothesis to $\vdash_{\mathrm{stack}} S : \mathcal{S} : \overline{\mathcal{S}}$ and $\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r' \succeq r_i$, we conclude that $\cdot, s \mapsto S^*; \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r' \succeq r_i\right]\!\right]\ [\tau_a]\ {\kappa^v}^* \hookrightarrow {\kappa_1^v}^*$ and $\cdot, s \mapsto S^*; \kappa_1^{v*} \hookrightarrow_\kappa S'^*; v'^*$.

Applying the induction hypothesis to $\vdash_{\mathrm{stack}} S : \mathcal{S} : \overline{\mathcal{S}}$ and $\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r \succeq r'$, we conclude that $\cdot, s \mapsto S^*; \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r \succeq r'\right]\!\right]\ [\tau_a]\ \kappa_1^{v*} \hookrightarrow \kappa_2^{v*}$ and $\cdot, s \mapsto S^*; \kappa_2^{v*} \hookrightarrow_\kappa S'^*; v'^*$.

Note that

$$\dfrac{\dfrac{\boxed{\cdot, s \mapsto S^*; v_w^*\ [\tau_a] \hookrightarrow \lambda k : \mathsf{RGN}\ s\sharp r_i\ \beta.\mathsf{let}\ k' = \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r' \succeq r_i\right]\!\right]\ [\tau_a]\ k\ \mathsf{in}\ \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r \succeq r'\right]\!\right]\ [\tau_a]\ k'}}{\cdot, s \mapsto S^*; {\kappa^v}^* \hookrightarrow {\kappa^v}^*}\quad \dfrac{\boxed{\cdot, s \mapsto S^*; \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r' \succeq r_i\right]\!\right]\ [\tau_a]\ {\kappa^v}^* \hookrightarrow \kappa_1^{v*}} \quad \boxed{\cdot, s \mapsto S^*; \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r \succeq r'\right]\!\right]\ [\tau_a]\ \kappa_1^{v*} \hookrightarrow \kappa_2^{v*}}}{\cdot, s \mapsto S^*; (\mathsf{let}\ k' = \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r' \succeq r_i\right]\!\right]\ [\tau_a]\ k\ \mathsf{in}\ \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r \succeq r'\right]\!\right]\ [\tau_a]\ k')[{\kappa^v}^*/k] \hookrightarrow \kappa_2^{v*}}}{\cdot, s \mapsto S^*; v_w^*\ [\tau_a]\ {\kappa^v}^* \hookrightarrow \kappa_2^{v*}}$$

and

$$\overline{\cdot, s \mapsto S^*; \kappa_2^{v^*} \hookrightarrow_\kappa S'^*; v'^*}$$

□

Lemma 41 (Translation Correctness (stack domain weakening))

Suppose $\overline{\mathcal{S}}' \supseteq_{sr} \overline{\mathcal{S}}$ *and* $\vdash_{\text{sdom}} \overline{\mathcal{S}'}$ *and* $\mathcal{S}' \supseteq_{sr} \mathcal{S}$ *and* $\vdash_{\text{stype}} \mathcal{S}' : \overline{\mathcal{S}}'$.
Suppose $\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta$.
Let $\mathbb{T}_e^{\vdash_{\text{exp}}} [\![\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta]\!] = e^*$.
Then there exists a derivation of $\Delta; \Gamma; \mathcal{S}' : \overline{\mathcal{S}}' \vdash_{\text{exp}} e : \tau, \theta$ *such that*
$e^* = \mathbb{T}_e^{\vdash_{\text{exp}}} \left[\!\!\left[\Delta; \Gamma; \mathcal{S}' : \overline{\mathcal{S}}' \vdash_{\text{exp}} e : \tau, \theta\right]\!\!\right]$.

Proof. Each judgement of the form $\overline{\mathcal{S}} \vdash \bigcirc$, $\Delta; \overline{\mathcal{S}} \vdash \bigcirc$, $\mathcal{S} : \overline{\mathcal{S}} \vdash \bigcirc$, and $\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash \bigcirc$ can be replaced by (syntactically) indentical judgements of the form $\overline{\mathcal{S}}' \vdash \bigcirc$, $\Delta; \overline{\mathcal{S}}' \vdash \bigcirc$, $\mathcal{S}' : \overline{\mathcal{S}}' \vdash \bigcirc$, and $\Delta; \Gamma; \mathcal{S}' : \overline{\mathcal{S}}' \vdash \bigcirc$ without invalidating any judgements. □

Lemma 42 (Translation Correctness (substitution of places))

Suppose $\Delta, \varrho \succeq \varphi, \Delta'; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta$ *and* $\Delta, \Delta'[\rho/\varrho]; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \varphi$.
Let $\mathbb{T}_e^{\vdash_{\text{exp}}} [\![\Delta, \varrho \succeq \varphi, \Delta'; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta]\!] = e^*$.
Then there exists a derivation of $\Delta, \Delta'[\rho/\varrho]; \Gamma[\rho/\varrho]; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e[\rho/\varrho] : \tau[\rho/\varrho], \theta[\rho/\varrho]$ *such that*
$\mathbb{T}_e^{\vdash_{\text{exp}}} [\![\Delta, \Delta'[\rho/\varrho]; \Gamma[\rho/\varrho]; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e[\rho/\varrho] : \tau[\rho/\varrho], \theta[\rho/\varrho]]\!] =$
$e^*[\mathbb{T}_\tau^\rho [\![\rho]\!] /\varrho][\mathbb{T}_v^{\vdash_{\text{re}}} [\![\Delta; \overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \varphi]\!] / w_\varrho][\mathbb{T}_v^{\vdash_{\text{place}}} [\![\Delta, \Delta'[\rho/\varrho]; \overline{\mathcal{S}} \vdash_{\text{place}} \rho]\!] / h_\varrho]$.

Lemma 43 (Translation Correctness (substitution of closed values))

Suppose $\Delta; \Gamma, x : \tau', \Gamma'; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta$ *and* $\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{cval}} v : \tau'$.
Let $\mathbb{T}_e^{\vdash_{\text{exp}}} [\![\Delta; \Gamma, x : \tau', \Gamma'; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta]\!] = e^*$.
Then there exists a derivation of $\Delta; \Gamma, \Gamma'; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e[v/x] : \tau, \theta$ *such that*
$\mathbb{T}_e^{\vdash_{\text{exp}}} [\![\Delta; \Gamma, \Gamma'; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e[v/x] : \tau, \theta]\!] = e^*[\mathbb{T}_v^{\vdash_{\text{cval}}} [\![\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{cval}} v : \tau']\!] /x]$.

Theorem 5 (Translation Correctness)

Suppose $\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}$, $\cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, r'$, *and* $S; e \hookrightarrow S'; v'$.
Then there exists $\overline{\mathcal{S}}' \supseteq_= \overline{\mathcal{S}}$ *and* $\mathcal{S}' \supseteq_= \mathcal{S}$ *such that* $\vdash_{\text{stack}} S' : \mathcal{S}' : \overline{\mathcal{S}}'$ *and* $\mathcal{S}' : \overline{\mathcal{S}}' \vdash_{\text{cval}} v' : \tau$.
Let $\mathbb{T}_{\overline{\mathcal{T}}}^{\vdash_{\text{sdom}}} [\![\vdash_{\text{sdom}} \overline{\mathcal{S}}]\!] = \cdot, s \mapsto \overline{\mathcal{S}}^*$, $\mathbb{T}_{\mathcal{T}}^{\vdash_{\text{stype}}} [\![\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}]\!] = \cdot, s \mapsto \mathcal{S}^*$, $\mathbb{T}_T^{\vdash_{\text{stack}}} [\![\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}]\!] = \cdot, s \mapsto S^*$,
and $\mathbb{T}_e^{\vdash_{\text{exp}}} [\![\cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, r']\!] = e^*$.
Then $\cdot, s \mapsto S^*; e^* \hookrightarrow \kappa^{v^{*\prime}}$ *and* $\cdot, s \mapsto S^*; \kappa^{v^{*\prime}} \hookrightarrow_\kappa S'^*; v'^*$,
where $\mathbb{T}_{\overline{\mathcal{S}}}^{\vdash_{\text{sdom}}} \left[\!\!\left[\vdash_{\text{sdom}} \overline{\mathcal{S}}'\right]\!\!\right] = \overline{\mathcal{S}}'^*$, $\mathbb{T}_{\mathcal{S}}^{\vdash_{\text{stype}}} \left[\!\!\left[\vdash_{\text{stype}} \mathcal{S}' : \overline{\mathcal{S}}'\right]\!\!\right] = \mathcal{S}'^*$, $\mathbb{T}_S^{\vdash_{\text{stack}}} \left[\!\!\left[\vdash_{\text{stack}} S' : \mathcal{S}' : \overline{\mathcal{S}}'\right]\!\!\right] = S'^*$, *and*
$\mathbb{T}_v^{\vdash_{\text{cval}}} \left[\!\!\left[\mathcal{S}' : \overline{\mathcal{S}}' \vdash_{\text{cval}} v' : \tau\right]\!\!\right] = v'^*$.

Proof. By applying Theorem 1 to $\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}$, $\cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, r'$, and $S; e \hookrightarrow S'; v'$, we conclude that there exists $\overline{\mathcal{S}}' \supseteq_= \overline{\mathcal{S}}$ and $\mathcal{S}' \supseteq_= \mathcal{S}$ such that $\vdash_{\text{stack}} S' : \mathcal{S}' : \overline{\mathcal{S}}'$ and $\mathcal{S}' : \overline{\mathcal{S}}' \vdash_{\text{cval}} v' : \tau$, as assumed.

By applying Lemma 9 to $\cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, r'$, we conclude that $\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} r'$. Hence, we conclude that $r' \in dom(\overline{\mathcal{S}})$. Furthermore, $r' \in dom(\mathcal{S})$ and $r' \in dom(S)$. Thus, $\overline{\mathcal{S}} \neq \cdot$, $\mathcal{S} \neq \cdot$, and $S \neq \cdot$. Hence, $\mathbb{T}_{\overline{\mathcal{T}}}^{\vdash_{\text{sdom}}} [\![\vdash_{\text{sdom}} \overline{\mathcal{S}}]\!] = \cdot, s \mapsto \mathbb{T}_{\overline{\mathcal{S}}}^{\vdash_{\text{sdom}}} [\![\vdash_{\text{sdom}} \overline{\mathcal{S}}]\!] = \cdot, s \mapsto \overline{\mathcal{S}}^*$, $\mathbb{T}_{\mathcal{T}}^{\vdash_{\text{stype}}} [\![\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}]\!] = \cdot, s \mapsto$

$\mathbb{T}_{\mathcal{S}}^{\vdash_{\text{stype}}}\left[\!\left[\vdash_{\text{stype}} \mathcal{S} : \overline{\mathcal{S}}\right]\!\right] = \cdot, s \mapsto \mathcal{S}^*$, and $\mathbb{T}_{T}^{\vdash_{\text{stack}}}\left[\!\left[\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}\right]\!\right] = \cdot, s \mapsto \mathbb{T}_{S}^{\vdash_{\text{stack}}}\left[\!\left[\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}\right]\!\right] = \cdot, s \mapsto S^*$ as assumed.

By applying Lemma 39 to $\cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, r'$, we conclude that $\cdot;\cdot;\cdot, s \mapsto \mathcal{S}^* : \cdot, s \mapsto \overline{\mathcal{S}}^* \vdash_{\text{exp}} e^* : \mathsf{RGN}\ s\sharp r'\ \mathbb{T}_{\tau}^{\vdash_{\text{type}}}\left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{type}} \tau\right]\!\right]$.

Proceed by induction on the derivation $S; e \hookrightarrow S'; v'$.

Case $\dfrac{\rho \equiv r \qquad r \in dom(S) \qquad l \notin dom(S(r))}{S; i\ \mathsf{at}\ \rho \hookrightarrow S\{(r,l) \mapsto i\}; \langle l \rangle_r}$: Note that

$$\frac{\vdash_{\text{ctxt}} \cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}}; r' \qquad \cdot;\overline{\mathcal{S}} \vdash_{\text{place}} r \qquad \cdot;\overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq r}{\cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} i\ \mathsf{at}\ r : (\mathsf{int}, r), r'}$$

and

$$\begin{array}{rcl} e^* & = & \mathbb{T}_{v}^{\vdash_{\text{rr}}}\left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq r\right]\!\right]\ [\mathbb{T}_{\tau}^{\vdash_{\text{type}}}\left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{type}} (\mathsf{int}, r)\right]\!\right]] \\ & & (\underline{\mathsf{newRGNVar}}\ [\mathbb{T}_{\tau}^{\vdash_{\text{place}}}\left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{place}} r\right]\!\right]] \\ & & \quad [\mathbb{T}_{\tau}^{\vdash_{\text{btype}}}\left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\right]] \\ & & \quad \mathbb{T}_{e}^{\vdash_{\text{place}}}\left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{place}} r\right]\!\right] \\ & & \quad i) \\ & = & \mathbb{T}_{v}^{\vdash_{\text{rr}}}\left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq r\right]\!\right]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ (\underline{\mathsf{newRGNVar}}\ [s\sharp r]\ [\mathsf{int}]\ \mathsf{handle}(s\sharp r)\ i) \end{array}$$

Note that

$$\frac{\boxed{\cdot;\cdot;\cdot, s \mapsto \mathcal{S}^* : \cdot, s \mapsto \overline{\mathcal{S}}^* \vdash_{\text{exp}} \mathsf{handle}(s\sharp r) : \mathsf{RGNHandle}\ s\sharp r} \qquad \boxed{\cdot;\cdot;\cdot, s \mapsto \mathcal{S}^* : \cdot, s \mapsto \overline{\mathcal{S}}^* \vdash_{\text{exp}} i : \mathsf{int}}}{\cdot;\cdot;\cdot, s \mapsto \mathcal{S}^* : \cdot, s \mapsto \overline{\mathcal{S}}^* \vdash_{\text{exp}} \underline{\mathsf{newRGNVar}}\ [s\sharp r]\ [\mathsf{int}]\ \mathsf{handle}(s\sharp r)\ i : \mathsf{RGN}\ s\sharp r\ \mathsf{int}}$$

and

$$\frac{\boxed{r \in dom(S^*)} \qquad \boxed{l \notin dom(S^*(r))}}{\cdot, s \mapsto S^*; \underline{\mathsf{newRGNVar}}\ [s\sharp r]\ [\mathsf{int}]\ \mathsf{handle}(s\sharp r)\ i \hookrightarrow_{\kappa} S^*\{(r,l) \mapsto i\}, \langle l \rangle_{s\sharp r}}$$

By Lemma 40, we conclude that

$$\begin{array}{c} \cdot, s \mapsto S^*; e^* \hookrightarrow \kappa^{v*\prime} \\ \text{and} \\ \cdot, s \mapsto S^*; \kappa^{v*\prime} \hookrightarrow_{\kappa} S^*\{(r,l) \mapsto i\}, \langle l \rangle_{s\sharp r} \end{array}$$

Note that

$$\begin{array}{rcl} S' & = & S\{(r,l) \mapsto i\} \\ \mathcal{S}' & = & \mathcal{S}\{(r,l) \mapsto \mathsf{int}\} \\ \overline{\mathcal{S}}' & = & \overline{\mathcal{S}}\{(r,l) \mapsto\} \end{array} \quad \text{and} \quad \vdash_{\text{stack}} S' : \mathcal{S}' : \overline{\mathcal{S}}' \quad \text{and} \quad \mathbb{T}_{S}^{\vdash_{\text{stack}}}\left[\!\left[\vdash_{\text{stack}} S' : \mathcal{S}' : \overline{\mathcal{S}}'\right]\!\right] = S^*\{(r,l) \mapsto i\}$$

Finally, note that $\mathbb{T}_{v}^{\vdash_{\text{cval}}}\left[\!\left[\mathcal{S}' : \overline{\mathcal{S}}' \vdash_{\text{cval}} \langle l \rangle_r : (\mathsf{int}, r)\right]\!\right] = \langle l \rangle_{s\sharp r}$.

Case $\dfrac{\begin{array}{c} S; e_1 \hookrightarrow S_1; v_1 \qquad v_1 \equiv \langle l_1 \rangle_{r_1} \\ r_1 \in dom(S_1) \qquad l_1 \in dom(S_1(r_1)) \qquad S_1(r_1, l_1) = i_1 \\ S_1; e_2 \hookrightarrow S_2; v_2 \qquad v_2 \equiv \langle l_2 \rangle_{r_2} \\ r_2 \in dom(S_2) \qquad l_2 \in dom(S_2(r_2)) \qquad S_2(r_2, l_2) = i_2 \\ \rho \equiv r \qquad r \in dom(S_2) \qquad l \notin dom(S_2(r)) \qquad i = i_1 \oplus i_2 \end{array}}{S; e_1 \oplus e_2\ \mathsf{at}\ \rho \hookrightarrow S_2\{(r,l) \mapsto i\}; \langle l \rangle_r}$: Note that

$$\frac{\begin{array}{c} \cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\mathsf{int}, \rho_1), r' \qquad \cdot;\overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq \rho_1 \\ \cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_2 : (\mathsf{int}, \rho_2), r' \qquad \cdot;\overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq \rho_2 \\ \cdot;\overline{\mathcal{S}} \vdash_{\text{place}} r \qquad \cdot;\overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq r \end{array}}{\cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 \oplus e_2\ \mathsf{at}\ r : (\mathsf{int}, r), r'}$$

Applying Theorem 1 to (1a) $\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}$, (1b) $\cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\mathsf{int}, \rho_1), r'$, and (1c) $S; e_1 \hookrightarrow S_1; \langle l_1 \rangle_{r_1}$, we conclude that there exists $\overline{\mathcal{S}}_1 \supseteq_{=\supseteq} \overline{\mathcal{S}}$ and $\mathcal{S}_1 \supseteq_{=\supseteq} \mathcal{S}$ such that $\vdash_{\text{stack}} S_1 : \mathcal{S}_1 : \overline{\mathcal{S}}_1$ and $\mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{cval}} \langle l_1 \rangle_{r_1} : (\mathsf{int}, \rho_1)$. Hence, $\rho_1 = r_1$.

By Lemma 13, we conclude $\cdot;\cdot;\mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{exp}} e_2 : (\mathsf{int}, \rho_2), r'$.

Applying Theorem 1 to (1a) $\vdash_{\text{stack}} S_1 : \mathcal{S}_1 : \overline{\mathcal{S}}_1$, (1b) $\cdot;\cdot;\mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{exp}} e_2 : (\mathsf{int}, \rho_2), r'$, and (1c) $S_1; e_2 \hookrightarrow S_2; \langle l_2 \rangle_{r_2}$, we conclude that there exists $\overline{\mathcal{S}}_2 \supseteq_{=\supseteq} \overline{\mathcal{S}}_1$ and $\mathcal{S}_2 \supseteq_{=\supseteq} \mathcal{S}_1$ such that $\vdash_{\text{stack}} S_2 : \mathcal{S}_2 : \overline{\mathcal{S}}_2$ and $\mathcal{S}_2 : \overline{\mathcal{S}}_2 \vdash_{\text{cval}} \langle l_2 \rangle_{r_2} : (\mathsf{int}, \rho_2)$. Hence, $\rho_2 = r_2$.

Note

$$
\begin{array}{rcl}
e^* & = & \mathsf{bind}\ a : \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} (\mathsf{int}, r_1)\right]\!\right] \Leftarrow \mathbb{T}_e^{\vdash \text{exp}} \left[\!\left[\cdot;\cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\mathsf{int}, r_1), r'\right]\!\right]; \\
& & \mathsf{bind}\ a' : \mathbb{T}_\tau^{\vdash \text{bype}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\right] \Leftarrow \mathbb{T}_v^{\vdash \text{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq r_1\right]\!\right] \ [\mathbb{T}_\tau^{\vdash \text{bype}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\right]] \\
& & \qquad (\underline{\mathsf{readRGNVar}}\ [\mathbb{T}_\tau^{\vdash \text{place}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} r_1\right]\!\right]] \\
& & \qquad\quad [\mathbb{T}_\tau^{\vdash \text{bype}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\right]] \\
& & \qquad\quad a); \\
& & \mathsf{bind}\ b : \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} (\mathsf{int}, r_2)\right]\!\right] \Leftarrow \mathbb{T}_e^{\vdash \text{exp}} \left[\!\left[\cdot;\cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_2 : (\mathsf{int}, r_2), r'\right]\!\right]; \\
& & \mathsf{bind}\ b' : \mathbb{T}_\tau^{\vdash \text{bype}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\right] \Leftarrow \mathbb{T}_v^{\vdash \text{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq r_2\right]\!\right] \ [\mathbb{T}_\tau^{\vdash \text{bype}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\right]] \\
& & \qquad (\underline{\mathsf{readRGNVar}}\ [\mathbb{T}_\tau^{\vdash \text{place}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} r_2\right]\!\right]] \\
& & \qquad\quad [\mathbb{T}_\tau^{\vdash \text{bype}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\right]] \\
& & \qquad\quad b); \\
& & \mathsf{let}\ z = a' \oplus b'\ \mathsf{in} \\
& & \mathbb{T}_v^{\vdash \text{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq r\right]\!\right] \ [\mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} (\mathsf{int}, r)\right]\!\right]] \\
& & \qquad (\underline{\mathsf{newRGNVar}}\ [\mathbb{T}_\tau^{\vdash \text{place}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} r\right]\!\right]] \\
& & \qquad\quad [\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{btype}} \mathsf{int}\right]\!\right]] \\
& & \qquad\quad \mathbb{T}_e^{\vdash \text{place}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} r\right]\!\right] \\
& & \qquad\quad z) \\
& & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{where } a, a', b, b', z \text{ fresh} \\
= & & \mathsf{bind}\ a : \mathsf{RGN}\ s\sharp r_1\ \mathsf{int} \Leftarrow \mathbb{T}_e^{\vdash \text{exp}} \left[\!\left[\cdot;\cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\mathsf{int}, r_1), r'\right]\!\right]; \\
& & \mathsf{bind}\ a' : \mathsf{int} \Leftarrow \mathbb{T}_v^{\vdash \text{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq r_1\right]\!\right] \ [\mathsf{int}]\ (\underline{\mathsf{readRGNVar}}\ [s\sharp r_1]\ [\mathsf{int}]\ a) \\
& & \mathsf{bind}\ b : \mathsf{RGN}\ s\sharp r_2\ \mathsf{int} \Leftarrow \mathbb{T}_e^{\vdash \text{exp}} \left[\!\left[\cdot;\cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_2 : (\mathsf{int}, r_2), r'\right]\!\right]; \\
& & \mathsf{bind}\ b' : \mathsf{int} \Leftarrow \mathbb{T}_v^{\vdash \text{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq r_2\right]\!\right] \ [\mathsf{int}]\ (\underline{\mathsf{readRGNVar}}\ [s\sharp r_2]\ [\mathsf{int}]\ b); \\
& & \mathsf{let}\ z = a' \oplus b'\ \mathsf{in} \\
& & \mathbb{T}_v^{\vdash \text{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq r\right]\!\right] \ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ (\underline{\mathsf{newRGNVar}}\ [s\sharp r]\ [\mathsf{int}]\ \mathsf{handle}(s\sharp r)\ z)
\end{array}
$$

By applying the induction hypothesis to $S; e_1 \hookrightarrow S_1; \langle l_1 \rangle_{r_1}$, $\cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\mathsf{int}, r_1), r'$, $\mathbb{T}_e^{\vdash \text{exp}} \left[\!\left[\cdot;\cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_1 : (\mathsf{int}, r_1), r'\right]\!\right] = e_1^*$, we conclude that there exists $\overline{\mathcal{S}}_1 \supseteq_{=\supseteq} \overline{\mathcal{S}}$ and $\mathcal{S}_1 \supseteq_{=\supseteq} \mathcal{S}$ such that $\vdash_{\text{stack}} S_1 : \mathcal{S}_1 : \overline{\mathcal{S}}_1$ and $\mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{cval}} \langle l_1 \rangle_{r_1} : \mathsf{int}$, and $\cdot, s \mapsto S^*; e_1^* \hookrightarrow \kappa_a^{v*}$ and $\cdot, s \mapsto S^*; \kappa_a^{v*} \hookrightarrow_\kappa S_1^*; v_1^*$ where $\mathbb{T}_S^{\vdash \text{stack}} \left[\!\left[\vdash_{\text{stack}} S_1 : \mathcal{S}_1 : \overline{\mathcal{S}}_1\right]\!\right] = S_1^*$, and $\mathbb{T}_v^{\vdash \text{cval}} \left[\!\left[\mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\text{cval}} \langle l_1 \rangle_{r_1} : (\mathsf{int}, r_1)\right]\!\right] = \langle l_1 \rangle_{s\sharp r_1} = {v_1}^*$.

Note that

$$
\begin{array}{rcl}
e^* & = & \mathsf{let}\ k_a = e_1^*\ \mathsf{in} \\
& & \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{RGNVar}\ s\sharp r_1\ \mathsf{int}]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ k_a\ (\lambda a : \mathsf{RGNVar}\ s\sharp r_1\ \mathsf{int}. e_a^*)
\end{array}
$$

where

$$
\begin{array}{rcl}
e_a^* & = & \mathsf{bind}\ a' : \mathsf{int} \Leftarrow \mathbb{T}_v^{\vdash \text{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq r_1\right]\!\right] \ [\mathsf{int}]\ (\underline{\mathsf{readRGNVar}}\ [s\sharp r_1]\ [\mathsf{int}]\ a); \\
& & \mathsf{bind}\ b : \mathsf{RGNVar}\ s\sharp r_2\ \mathsf{int} \Leftarrow \mathbb{T}_e^{\vdash \text{exp}} \left[\!\left[\cdot;\cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e_2 : (\mathsf{int}, r_2), r'\right]\!\right]; \\
& & \mathsf{bind}\ b' : \mathsf{int} \Leftarrow \mathbb{T}_v^{\vdash \text{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq r_2\right]\!\right] \ [\mathsf{int}]\ (\underline{\mathsf{readRGNVar}}\ [s\sharp r_2]\ [\mathsf{int}]\ b); \\
& & \mathsf{let}\ z = a' \oplus b'\ \mathsf{in} \\
& & \mathbb{T}_v^{\vdash \text{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq r\right]\!\right] \ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ (\underline{\mathsf{newRGNVar}}\ [s\sharp r]\ [\mathsf{int}]\ \mathsf{handle}(s\sharp r)\ z)
\end{array}
$$

Hence,

$$
\dfrac{\dfrac{}{\cdot, s \mapsto S^*; e_1^* \hookrightarrow \kappa_a^{v*}}}{\cdot, s \mapsto S^*; \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{RGNVar}\ s\sharp r_1\ \mathsf{int}]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ \kappa_a^{v*}\ (\lambda a : \mathsf{RGNVar}\ s\sharp r_1\ \mathsf{int}. e_a^*) \hookrightarrow \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{RGNVar}\ s\sharp r_1\ \mathsf{int}]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ \kappa_a^{v*}\ (\lambda a : \mathsf{RGNVar}\ s\sharp r_1\ \mathsf{int}. e_a^*)}{\cdot, s \mapsto S^*; e^* \hookrightarrow \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{RGNVar}\ s\sharp r_1\ \mathsf{int}]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ \kappa_a^{v*}\ (\lambda a : \mathsf{RGNVar}\ s\sharp r_1\ \mathsf{int}. e_a^*)}
$$

Note that

$$\frac{\boxed{\cdot, s \mapsto S^*; \kappa_a^{v\,*} \hookrightarrow_\kappa S_1^*; \langle l_1 \rangle_{s\sharp r_1}} \qquad \frac{\boxed{\cdot, s \mapsto S_1^*; e_a^*[\langle l_1 \rangle_{s\sharp r_1}/a] \hookrightarrow \kappa_A^{v\,*}}}{\cdot, s \mapsto S_1^*; (\lambda a : \mathsf{RGNVar}\ s\sharp r_1\ \mathsf{int}.e_a^*)\ \langle l_1 \rangle_{s\sharp r_1} \hookrightarrow \kappa_A^{v\,*}} \qquad \boxed{\cdot, s \mapsto S_1^*; \kappa_A^{v\,*} \hookrightarrow_\kappa \bigcirc}}{\cdot, s \mapsto S^*; \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{RGNVar}\ s\sharp r_1\ \mathsf{int}]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ \kappa_a^{v\,*}\ (\lambda a : \mathsf{RGNVar}\ s\sharp r_1\ \mathsf{int}.e_a^*) \hookrightarrow_\kappa \bigcirc}$$

Note that

$$\begin{array}{rcl} e_a^*[\langle l_1 \rangle_{s\sharp r_1}/a] & = & \mathsf{let}\ k_{a'} = \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r' \succeq r_1 \right]\!\right]\ [\mathsf{int}]\ (\underline{\mathsf{readRGNVar}}\ [s\sharp r_1]\ [\mathsf{int}]\ \langle l_1 \rangle_{s\sharp r_1})\ \mathsf{in} \\ & & \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{int}]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ k_{a'}\ (\lambda a' : \mathsf{int}.e_{a'}^*) \end{array}$$

where

$$\begin{array}{rcl} e_{a'}^* & = & \mathsf{bind}\ b : \mathsf{RGNVar}\ s\sharp r_2\ \mathsf{int} \Leftarrow \mathbb{T}_e^{\vdash \mathrm{exp}} \left[\!\left[\cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\mathrm{exp}} e_2 : (\mathsf{int}, r_2), r' \right]\!\right]; \\ & & \mathsf{bind}\ b' : \mathsf{int} \Leftarrow \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r' \succeq r_2 \right]\!\right]\ [\mathsf{int}]\ (\underline{\mathsf{readRGNVar}}\ [s\sharp r_2]\ [\mathsf{int}]\ b); \\ & & \mathsf{let}\ z = a' \oplus b'\ \mathsf{in} \\ & & \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r' \succeq r \right]\!\right]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ (\underline{\mathsf{newRGNVar}}\ [s\sharp r]\ [\mathsf{int}]\ \mathsf{handle}(s\sharp r)\ z) \end{array}$$

Hence

$$\frac{\boxed{\cdot, s \mapsto S_1^*; \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r' \succeq r_1 \right]\!\right]\ [\mathsf{int}]\ (\underline{\mathsf{readRGNVar}}\ [s\sharp r_1]\ [\mathsf{int}]\ \langle l_1 \rangle_{s\sharp r_1}) \hookrightarrow \kappa_{a'}^{v\,*}} \qquad \boxed{\begin{array}{c} \cdot, s \mapsto S_1^*; \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{int}]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ \kappa_{a'}^{v\,*}\ (\lambda a' : \mathsf{int}.e_{a'}^*) \hookrightarrow \\ \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{int}]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ \kappa_{a'}^{v\,*}\ (\lambda a' : \mathsf{int}.e_{a'}^*) \end{array}}}{\cdot, s \mapsto S_1^*; e_a^*[\langle l_1 \rangle_{s\sharp r_1}/a] \hookrightarrow \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{int}]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ \kappa_{a'}^{v\,*}\ (\lambda a' : \mathsf{int}.e_{a'}^*)}$$

Note that

$$\frac{\boxed{\cdot; \cdot; \cdot, s \mapsto \mathcal{S}_1^* : \cdot, s \mapsto \overline{\mathcal{S}}_1^* \vdash_{\mathrm{exp}} \langle l_1 \rangle_{s\sharp r_1} : \mathsf{RGNVar}\ s\sharp r_1\ \mathsf{int}}}{\cdot; \cdot; \cdot, s \mapsto \mathcal{S}_1^* : \cdot, s \mapsto \overline{\mathcal{S}}_1^* \vdash_{\mathrm{exp}} \underline{\mathsf{readRGNVar}}\ [s\sharp r_1]\ [\mathsf{int}]\ \langle l_1 \rangle_{s\sharp r_1} : \mathsf{RGN}\ s\sharp r_1\ \mathsf{int}}$$

and

$$\frac{\boxed{r_1 \in dom(S_1^*)} \qquad \boxed{l_1 \in dom(S_1^*(r_1))} \qquad \boxed{i_1 = S_1^*(r_1, l_1)}}{\cdot, s \mapsto S_1^*; \underline{\mathsf{readRGNVar}}\ [s\sharp r_1]\ [\mathsf{int}]\ \langle l_1 \rangle_{s\sharp r_1} \hookrightarrow_\kappa S_1^*; i_1}$$

By Lemma 40, we conclude that

$$\begin{array}{c} \cdot, s \mapsto S_1^*; \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r' \succeq r_1 \right]\!\right]\ [\mathsf{int}]\ (\underline{\mathsf{readRGNVar}}\ [s\sharp r_1]\ [\mathsf{int}]\ \langle l_1 \rangle_{s\sharp r_1}) \hookrightarrow \kappa_{a'}^{v\,*} \\ \text{and} \\ \cdot, s \mapsto S_1^*; \kappa_{a'}^{v\,*} \hookrightarrow_\kappa S_1^*; i_1 \end{array}$$

Note that

$$\frac{\boxed{\cdot, s \mapsto S_1^*; \kappa_{a'}^{v\,*} \hookrightarrow_\kappa S_1^*; i_1} \qquad \frac{\boxed{\cdot, s \mapsto S_1^*; e_{a'}^*[i_1/a'] \hookrightarrow \kappa_{A'}^{v\,*}}}{\cdot, s \mapsto S_1^*; (\lambda a' : \mathsf{int}.e_{a'}^*)\ i_1 \hookrightarrow \kappa_{A'}^{v\,*}} \qquad \boxed{\cdot, s \mapsto S_1^*; \kappa_{A'}^{v\,*} \hookrightarrow_\kappa \bigcirc}}{\cdot, s \mapsto S_1^*; \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{int}]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ \kappa_{a'}^{v\,*}\ (\lambda a' : \mathsf{int}.e_{a'}^*) \hookrightarrow_\kappa \bigcirc}$$

By applying the induction hypothesis to $S_1; e_2 \hookrightarrow S_2; \langle l_2 \rangle_{r_2}$, $\cdot; \cdot; \mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\mathrm{exp}} e_2 : (\mathsf{int}, r_2), r'$, $\mathbb{T}_e^{\vdash \mathrm{exp}} \left[\!\left[\cdot; \cdot; \mathcal{S}_1 : \overline{\mathcal{S}}_1 \vdash_{\mathrm{exp}} e_2 : (\mathsf{int}, r_2), r' \right]\!\right] = e_2^*$, we conclude that there exists $\overline{\mathcal{S}}_2 \supseteq_{=\supseteq} \overline{\mathcal{S}}_1$ and $\mathcal{S}_2 \supseteq_{=\supseteq} \mathcal{S}_1$ such that $\vdash_{\mathrm{stack}} S_2 : \mathcal{S}_2 : \overline{\mathcal{S}}_2$ and $\cdot; \cdot; \mathcal{S}_2 : \overline{\mathcal{S}}_2 \vdash_{\mathrm{cval}} \langle l_2 \rangle_{r_2} : \mathsf{int}$, and $\cdot, s \mapsto S_1^*; e_2^* \hookrightarrow \kappa_b^{v*}$ and $\cdot, s \mapsto S_1^*; \kappa_b^{v*\prime} \hookrightarrow_\kappa S_2^*; v_2$ where $\mathbb{T}_S^{\vdash \mathrm{stack}} \left[\!\left[\vdash_{\mathrm{stack}} S_2 : \mathcal{S}_2 : \overline{\mathcal{S}}_2 \right]\!\right] = S_2^*$, and $\mathbb{T}_v^{\vdash \mathrm{cval}} \left[\!\left[\cdot; \cdot; \mathcal{S}_2 : \overline{\mathcal{S}}_2 \vdash_{\mathrm{cval}} \langle l_2 \rangle_{r_2} : (\mathsf{int}, r_2) \right]\!\right] = \langle l_2 \rangle_{s\sharp r_2}$.
Note that

$$\begin{array}{rcl} e_{a'}^*[i_1/a'] & = & \mathsf{let}\ k_b = e_2^*\ \mathsf{in} \\ & & \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{RGNVar}\ s\sharp r_2\ \mathsf{int}]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ k_b\ (\lambda b : \mathsf{RGNVar}\ s\sharp r_2\ \mathsf{int}.e_b^*) \end{array}$$

where

$$\begin{array}{rcl} e_b^* & = & \mathsf{bind}\ b' : \mathsf{int} \Leftarrow \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r' \succeq r_2 \right]\!\right]\ [\mathsf{int}]\ (\underline{\mathsf{readRGNVar}}\ [s\sharp r_2]\ [\mathsf{int}]\ b); \\ & & \mathsf{let}\ z = i_1 \oplus b'\ \mathsf{in} \\ & & \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{rr}} r' \succeq r \right]\!\right]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ (\underline{\mathsf{newRGNVar}}\ [s\sharp r]\ [\mathsf{int}]\ \mathsf{handle}(s\sharp r)\ z) \end{array}$$

Hence,

$$\dfrac{\boxed{\cdot, s \mapsto S_1^*; e_2^* \hookrightarrow \kappa_b^{v*}}}{\dfrac{\boxed{\cdot, s \mapsto S_1^*; \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{RGNVar}\ s\sharp r_2\ \mathsf{int}]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ \kappa_b^{v*}\ (\lambda b : \mathsf{RGNVar}\ s\sharp r_2\ \mathsf{int}.e_b^*) \hookrightarrow \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{RGNVar}\ s\sharp r_2\ \mathsf{int}]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ \kappa_b^{v*}\ (\lambda b : \mathsf{RGNVar}\ s\sharp r_2\ \mathsf{int}.e_b^*)}}{\cdot, s \mapsto S_1^*; e_{a'}^*[i_1/a'] \hookrightarrow \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{RGNVar}\ s\sharp r_2\ \mathsf{int}]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ \kappa_b^{v*}\ (\lambda b : \mathsf{RGNVar}\ s\sharp r_2\ \mathsf{int}.e_b^*)}$$

Note that

$$\dfrac{\boxed{\cdot, s \mapsto S_1^*; \kappa_b^{v*} \hookrightarrow_\kappa S_2^*; \langle l_2\rangle_{s\sharp r_2}} \quad \dfrac{\boxed{\cdot, s \mapsto S_2^*; e_b^*[\langle l_2\rangle_{s\sharp r_2}/b] \hookrightarrow \kappa_B^{v*}}}{\cdot, s \mapsto S_2^*; (\lambda b : \mathsf{RGNVar}\ s\sharp r_2\ \mathsf{int}.e_b^*)\ \langle l_2\rangle_{s\sharp r_2} \hookrightarrow \kappa_B^{v*}} \quad \boxed{\cdot, s \mapsto S_2^*; \kappa_B^{v*} \hookrightarrow_\kappa \bigcirc}}{\cdot, s \mapsto S_1^*; \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{RGNVar}\ s\sharp r_2\ \mathsf{int}]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ \kappa_b^{v*}\ (\lambda b : \mathsf{RGNVar}\ s\sharp r_2\ \mathsf{int}.e_b^*) \hookrightarrow_\kappa \bigcirc}$$

Note that

$$\begin{array}{rcl} e_b^*[\langle l_2\rangle_{s\sharp r_2}/b] & = & \mathsf{let}\ k_{b'} = \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathbb{S}} \vdash_{\mathrm{rr}} r' \succeq r_2 \right]\!\right]\ [\mathsf{int}]\ (\underline{\mathsf{readRGNVar}}\ [s\sharp r_2]\ [\mathsf{int}]\ \langle l_2\rangle_{s\sharp r_2})\ \mathsf{in} \\ & & \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{int}]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ k_{b'}\ (\lambda b' : \mathsf{int}.e_{b'}^*) \end{array}$$

where

$$\begin{array}{rcl} e_{b'}^* & = & \mathsf{let}\ z = i_1 \oplus b'\ \mathsf{in} \\ & & \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathbb{S}} \vdash_{\mathrm{rr}} r' \succeq r \right]\!\right]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ (\underline{\mathsf{newRGNVar}}\ [s\sharp r]\ [\mathsf{int}]\ \mathsf{handle}(s\sharp r)\ z) \end{array}$$

Hence

$$\dfrac{\boxed{\cdot, s \mapsto S_2^*; \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathbb{S}} \vdash_{\mathrm{rr}} r' \succeq r_2 \right]\!\right]\ [\mathsf{int}]\ (\underline{\mathsf{readRGNVar}}\ [s\sharp r_2]\ [\mathsf{int}]\ \langle l_2\rangle_{s\sharp r_2}) \hookrightarrow \kappa_{b'}^{v*}} \quad \boxed{\begin{array}{l}\cdot, s \mapsto S_2^*; \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{int}]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ \kappa_{b'}^{v*}\ (\lambda b' : \mathsf{int}.e_{b'}^*) \hookrightarrow \\ \qquad \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{int}]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ \kappa_{b'}^{v*}\ (\lambda b' : \mathsf{int}.e_{b'}^*)\end{array}}}{\cdot, s \mapsto S_2^*; e_b^*[\langle l_2\rangle_{s\sharp r_2}/b] \hookrightarrow \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{int}]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ \kappa_{b'}^{v*}\ (\lambda b' : \mathsf{int}.e_{b'}^*)}$$

Note that

$$\dfrac{\boxed{\cdot; \cdot; \cdot, s \mapsto \mathbb{S}_2^* : \cdot, s \mapsto \overline{\mathbb{S}}_2^* \vdash_{\mathrm{exp}} \langle l_2\rangle_{s\sharp r_2} : \mathsf{RGNVar}\ s\sharp r_2\ \mathsf{int}}}{\cdot; \cdot; \cdot, s \mapsto \mathbb{S}_2^* : \cdot, s \mapsto \overline{\mathbb{S}}_2^* \vdash_{\mathrm{exp}} \underline{\mathsf{readRGNVar}}\ [s\sharp r_2]\ [\mathsf{int}]\ \langle l_2\rangle_{s\sharp r_2} : \mathsf{RGN}\ s\sharp r_2\ \mathsf{int}}$$

and

$$\dfrac{\boxed{r_2 \in dom(S_2^*)} \quad \boxed{l_2 \in dom(S_2^*(r_2))} \quad \boxed{i_2 = S_2^*(r_2, l_2)}}{\cdot, s \mapsto S_2^*; \underline{\mathsf{readRGNVar}}\ [s\sharp r_2]\ [\mathsf{int}]\ \langle l_2\rangle_{s\sharp r_2} \hookrightarrow_\kappa S_2^*; i_2}$$

By Lemma 40, we conclude that

$$\begin{array}{c} \cdot, s \mapsto S_2^*; \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathbb{S}} \vdash_{\mathrm{rr}} r' \succeq r_2 \right]\!\right]\ [\mathsf{int}]\ (\underline{\mathsf{readRGNVar}}\ [s\sharp r_2]\ [\mathsf{int}]\ \langle l_2\rangle_{s\sharp r_2}) \hookrightarrow \kappa_{b'}^{v*} \\ \text{and} \\ \cdot, s \mapsto S_2^*; \kappa_{b'}^{v*} \hookrightarrow_\kappa S_2^*; i_2 \end{array}$$

Note that

$$\dfrac{\boxed{\cdot, s \mapsto S_2^*; \kappa_{b'}^{v*} \hookrightarrow_\kappa S_2^*; i_2} \quad \dfrac{\boxed{\cdot, s \mapsto S_2^*; e_{b'}^*[i_2/b'] \hookrightarrow \kappa_{B'}^{v*}}}{\cdot, s \mapsto S_2^*; (\lambda b' : \mathsf{int}.e_{b'}^*)\ i_2 \hookrightarrow \kappa_{B'}^{v*}} \quad \boxed{\cdot, s \mapsto S_2^*; \kappa_{B'}^{v*} \hookrightarrow_\kappa \bigcirc}}{\cdot, s \mapsto S_2^*; \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{int}]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ \kappa_{b'}^{v*}\ (\lambda b' : \mathsf{int}.e_{b'}^*) \hookrightarrow_\kappa \bigcirc}$$

Note that

$$\begin{array}{rcl} e_{b'}^*[i_2/b'] & = & \mathsf{let}\ z = i_1 \oplus i_2\ \mathsf{in} \\ & & \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathbb{S}} \vdash_{\mathrm{rr}} r' \succeq r \right]\!\right]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ (\underline{\mathsf{newRGNVar}}\ [s\sharp r]\ [\mathsf{int}]\ \mathsf{handle}(s\sharp r)\ z) \end{array}$$

Hence,

$$\dfrac{\dfrac{\boxed{\cdot, s \mapsto S_2^*; i_1 \hookrightarrow i_1} \quad \boxed{\cdot, s \mapsto S_2^*; i_2 \hookrightarrow i_2} \quad \boxed{i = i_1 \oplus i_2}}{\cdot, s \mapsto S_2^*; i_1 \oplus i_2 \hookrightarrow i} \quad \boxed{\cdot, s \mapsto S_2^*; \mathbb{T}_v^{\vdash \mathrm{rr}} \left[\!\left[\cdot; \overline{\mathbb{S}} \vdash_{\mathrm{rr}} r' \succeq r \right]\!\right]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ (\underline{\mathsf{newRGNVar}}\ [s\sharp r]\ [\mathsf{int}]\ \mathsf{handle}(s\sharp r)\ i) \hookrightarrow \kappa_{B'}^{v*}}}{\cdot, s \mapsto S_2^*; e_{b'}^*[i_2/b'] \hookrightarrow \kappa_{B'}^{v*}}$$

Note that

$$\frac{\boxed{\cdot;\cdot;\cdot, s \mapsto \mathbb{S}_2^* : \cdot, s \mapsto \overline{\mathbb{S}}_2^* \vdash_{\text{exp}} \mathsf{handle}(s\sharp r) : \mathsf{RGNHandle}\ s\sharp r} \qquad \boxed{\cdot;\cdot;\cdot, s \mapsto \mathbb{S}_2^* : \cdot, s \mapsto \overline{\mathbb{S}}_2^* \vdash_{\text{exp}} i : \mathsf{int}}}{\cdot;\cdot;\cdot, s \mapsto \mathbb{S}_2^* : \cdot, s \mapsto \overline{\mathbb{S}}_2^* \vdash_{\text{exp}} \underline{\mathsf{newRGNVar}}\ [s\sharp r]\ [\mathsf{int}]\ \mathsf{handle}(s\sharp r)\ i : \mathsf{RGN}\ \mathbb{T}_\tau^\rho [\![s\sharp r]\!]\ \mathsf{int}}$$

and

$$\frac{\boxed{r \in dom(S_2^*)} \qquad \boxed{l \notin dom(S_2^*(r))}}{\cdot, s \mapsto S_2^*; \underline{\mathsf{newRGNVar}}\ [s\sharp r]\ [\mathsf{int}]\ \mathsf{handle}(s\sharp r)\ i \hookrightarrow_\kappa S_2^*\{(r,l) \mapsto i\}, \langle l \rangle_{s\sharp r}}$$

By Lemma 40, we conclude that

$$\cdot, s \mapsto S_2^*; \mathbb{T}_v^{\vdash \text{rr}} \left[\!\!\left[\cdot; \overline{\mathbb{S}} \vdash_{\text{rr}} r' \succeq r \right]\!\!\right]\ [\mathsf{RGNVar}\ s\sharp r\ \mathsf{int}]\ (\underline{\mathsf{newRGNVar}}\ [s\sharp r]\ [\mathsf{int}]\ \mathsf{handle}(s\sharp r)\ i) \hookrightarrow \kappa_{B'}^{v\ *}$$
$$\text{and}$$
$$\cdot, s \mapsto S_2^*; \kappa_{B'}^{v\ *} \hookrightarrow_\kappa S_2^*\{(r,l) \mapsto i\}, \langle l \rangle_{s\sharp r}$$

Note that

$$\begin{array}{lcl} S' & = & S_2\{(r,l) \mapsto i\} \\ \mathbb{S}' & = & \mathbb{S}_2\{(r,l) \mapsto \mathsf{int}\} \\ \overline{\mathbb{S}}' & = & \overline{\mathbb{S}}_2\{(r,l) \mapsto\} \end{array} \quad \text{and} \quad \vdash_{\text{stack}} S' : \mathbb{S}' : \overline{\mathbb{S}}' \quad \text{and} \quad \mathbb{T}_S^{\vdash \text{stack}} \left[\!\!\left[\vdash_{\text{stack}} S' : \mathbb{S}' : \overline{\mathbb{S}}' \right]\!\!\right] = S_2^*\{(r,l) \mapsto i\}$$

Finally, note that $\mathbb{T}_v^{\vdash \text{cval}} \left[\!\!\left[\cdot;\cdot; \mathbb{S}' : \overline{\mathbb{S}}' \vdash_{\text{cval}} \langle l \rangle_r : (\mathsf{int}, r) \right]\!\!\right] = \langle l \rangle_{s\sharp r}$.

Case $\dfrac{\begin{array}{c} S; e_1 \hookrightarrow S_1; v_1 \qquad v_1 \equiv \langle l_1 \rangle_{r_1} \\ r_1 \in dom(S_1) \qquad l_1 \in dom(S_1(r_1)) \qquad S_1(r_1, l_1) = i_1 \\ S_1; e_2 \hookrightarrow S_2; v_2 \qquad v_2 \equiv \langle l_2 \rangle_{r_2} \\ r_2 \in dom(S_2) \qquad l_2 \in dom(S_2(r_2)) \qquad S_2(r_2, l_2) = i_2 \\ b = i_1 \otimes i_2 \end{array}}{S; e_1 \otimes e_2\ \mathsf{at}\ \varrho \hookrightarrow S_2; b}$: ...

Case $\dfrac{}{S; \mathsf{tt} \hookrightarrow S; \mathsf{tt}}$: Note that

$$\frac{\vdash_{\text{ctxt}} \cdot;\cdot; \mathbb{S} : \overline{\mathbb{S}}; r'}{\cdot;\cdot; \mathbb{S} : \overline{\mathbb{S}} \vdash_{\text{exp}} \mathsf{tt} : \mathsf{bool}, r'}$$

and

$$\begin{array}{lcl} e^* & = & \underline{\mathsf{returnRGN}}\ [\mathbb{T}_\tau^{\vdash \text{place}} \left[\!\!\left[\cdot; \overline{\mathbb{S}} \vdash_{\text{place}} r' \right]\!\!\right]]\ [\mathbb{T}_\tau^{\vdash \text{type}} \left[\!\!\left[\cdot; \overline{\mathbb{S}} \vdash_{\text{type}} \mathsf{bool} \right]\!\!\right]]\ \mathsf{tt} \\ & = & \underline{\mathsf{returnRGN}}\ [s\sharp r']\ [\mathsf{bool}]\ \mathsf{tt} \end{array}$$

Note that

$$\frac{}{\cdot, s \mapsto S^*; \underline{\mathsf{returnRGN}}\ [s\sharp r']\ [\mathsf{bool}]\ \mathsf{tt} \hookrightarrow \underline{\mathsf{returnRGN}}\ [s\sharp r']\ [\mathsf{bool}]\ \mathsf{tt}}$$

and

$$\frac{}{\cdot, s \mapsto S^*; \underline{\mathsf{returnRGN}}\ [s\sharp r']\ [\mathsf{bool}]\ \mathsf{tt} \hookrightarrow_\kappa S^*; \mathsf{tt}}$$

Note that

$$\begin{array}{lcl} S' & = & S \\ \mathbb{S}' & = & \mathbb{S} \\ \overline{\mathbb{S}}' & = & \overline{\mathbb{S}} \end{array} \quad \text{and} \quad \vdash_{\text{stack}} S' : \mathbb{S}' : \overline{\mathbb{S}}' \quad \text{and} \quad \mathbb{T}_S^{\vdash \text{stack}} \left[\!\!\left[\vdash_{\text{stack}} S' : \mathbb{S}' : \overline{\mathbb{S}}' \right]\!\!\right] = S^*$$

Finally, note that $\mathbb{T}_v^{\vdash \text{cval}} \left[\!\!\left[\cdot;\cdot; \mathbb{S}' : \overline{\mathbb{S}}' \vdash_{\text{cval}} \mathsf{tt} : \mathsf{bool} \right]\!\!\right] = \mathsf{tt}$.

Case $\dfrac{}{S; \mathsf{ff} \hookrightarrow S; \mathsf{ff}}$: ...

Case $\dfrac{S; e_b \hookrightarrow S'; v' \qquad v' \equiv \mathsf{tt} \qquad S'; e_t \hookrightarrow S''; v''}{S; \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f \hookrightarrow S''; v''}$: ...

Case $\dfrac{S; e_b \hookrightarrow S'; v' \qquad v' \equiv \mathsf{ff} \qquad S'; e_f \hookrightarrow S''; v''}{S; \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f \hookrightarrow S''; v''}$: ...

Case $\dfrac{\rho \equiv r \qquad r \in dom(S) \qquad l \notin dom(S(r))}{S; \lambda x : \tau.^{\theta'} e' \text{ at } \rho \hookrightarrow S\{(r,l) \mapsto \lambda x : \tau.^{\theta'} e'\}; \langle l \rangle_r}$: Note that

$$\dfrac{\cdot; \cdot, x : \tau_1; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e' : \tau_2, \theta' \quad \cdot; \overline{\mathcal{S}} \vdash_{\text{place}} r \quad \cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq r}{\cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \lambda x : \tau_1.^{\theta'} e' \text{ at } r : (\tau_1 \xrightarrow{\theta'} \tau_2, r), r'}$$

and

$$\begin{array}{rcl}
e^* & = & \mathbb{T}_v^{\vdash \text{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq r \right]\!\right] \; [\mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} (\tau_1 \xrightarrow{\theta'} \tau_2, r) \right]\!\right]] \\
& & (\underline{\mathsf{newRGNVar}} \; [\mathbb{T}_\tau^{\vdash \text{place}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} r \right]\!\right]] \\
& & \quad [\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{btype}} \tau_1 \xrightarrow{\theta'} \tau_2 \right]\!\right]] \\
& & \quad \mathbb{T}_e^{\vdash \text{place}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} r \right]\!\right] \\
& & \quad (\lambda x : \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_1 \right]\!\right] . \mathbb{T}_e^{\vdash \text{exp}} \left[\!\left[\cdot; \cdot, x : \tau_1; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e' : \tau_2, \theta' \right]\!\right])) \\
& = & \mathbb{T}_v^{\vdash \text{rr}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq r \right]\!\right] \; [\mathsf{RGNVar}\ s\sharp r \; (\mathbb{T}_\tau^{\vdash \text{type}} [\![\tau_1]\!] \longrightarrow \mathsf{RGN}\ \mathbb{T}_\tau^{\vdash \text{place}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} \theta' \right]\!\right] \; \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_2 \right]\!\right])] \\
& & (\underline{\mathsf{newRGNVar}} \; [s\sharp r] \\
& & \quad [\mathbb{T}_\tau^{\vdash \text{type}} [\![\tau_1]\!] \longrightarrow \mathsf{RGN}\ \mathbb{T}_\tau^{\vdash \text{place}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} \theta' \right]\!\right] \; \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_2 \right]\!\right]] \\
& & \quad \mathsf{handle}(s\sharp r) \\
& & \quad (\lambda x : \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_1 \right]\!\right] . \mathbb{T}_e^{\vdash \text{exp}} \left[\!\left[\cdot; \cdot, x : \tau_1; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e' : \tau_2, \theta' \right]\!\right]))
\end{array}$$

Note that

$$\dfrac{\dfrac{\boxed{\cdot; \cdot; \cdot, s \mapsto \mathcal{S}^* : \cdot, s \mapsto \overline{\mathcal{S}}^* \vdash_{\text{exp}} \mathsf{handle}(s\sharp r) : \mathsf{RGNHandle}\ s\sharp r}}{\dfrac{\boxed{\begin{array}{l}\cdot; \cdot, x : \mathbb{T}_\tau^{\vdash \text{type}} [\![\tau_1]\!]; \cdot, s \mapsto \mathcal{S}^* : \cdot, s \mapsto \overline{\mathcal{S}}^* \vdash_{\text{exp}} \\ \quad \mathbb{T}_e^{\vdash \text{exp}} \left[\!\left[\cdot; \cdot, x : \tau_1; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e' : \tau_2, \theta' \right]\!\right] : \\ \quad \mathsf{RGN}\ \mathbb{T}_\tau^{\vdash \text{place}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} \theta' \right]\!\right] \; \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_2 \right]\!\right]\end{array}}}{\begin{array}{l}\cdot; \cdot; \cdot, s \mapsto \mathcal{S}^* : \cdot, s \mapsto \overline{\mathcal{S}}^* \vdash_{\text{exp}} \\ \quad \lambda x : \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_1 \right]\!\right] . \mathbb{T}_e^{\vdash \text{exp}} \left[\!\left[\cdot; \cdot, x : \tau_1; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e' : \tau_2, \theta' \right]\!\right] : \\ \quad \mathbb{T}_\tau^{\vdash \text{type}} [\![\tau_1]\!] \longrightarrow \mathsf{RGN}\ \mathbb{T}_\tau^{\vdash \text{place}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} \theta' \right]\!\right] \; \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_2 \right]\!\right]\end{array}}}{\begin{array}{l}\cdot; \cdot; \cdot, s \mapsto \mathcal{S}^* : \cdot, s \mapsto \overline{\mathcal{S}}^* \vdash_{\text{exp}} \\ \quad \underline{\mathsf{newRGNVar}} \; [s\sharp r] \\ \qquad [\mathbb{T}_\tau^{\vdash \text{type}} [\![\tau_1]\!] \longrightarrow \mathsf{RGN}\ \mathbb{T}_\tau^{\vdash \text{place}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} \theta' \right]\!\right] \; \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_2 \right]\!\right]] \\ \qquad \mathsf{handle}(s\sharp r) \\ \qquad (\lambda x : \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_1 \right]\!\right] . \mathbb{T}_e^{\vdash \text{exp}} \left[\!\left[\cdot; \cdot, x : \tau_1; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e' : \tau_2, \theta' \right]\!\right])) \\ \quad \mathsf{RGNVar}\ s\sharp r \; (\mathbb{T}_\tau^{\vdash \text{type}} [\![\tau_1]\!] \longrightarrow \mathsf{RGN}\ \mathbb{T}_\tau^{\vdash \text{place}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} \theta' \right]\!\right] \; \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_2 \right]\!\right])\end{array}} : }$$

and

$$\dfrac{\boxed{r \in dom(S^*)} \qquad \boxed{l \notin dom(S^*(r))}}{\begin{array}{l}\cdot, s \mapsto S^*; \underline{\mathsf{newRGNVar}} \; [s\sharp r] \\ \qquad [\mathbb{T}_\tau^{\vdash \text{type}} [\![\tau_1]\!] \longrightarrow \mathsf{RGN}\ \mathbb{T}_\tau^{\vdash \text{place}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} \theta' \right]\!\right] \; \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_2 \right]\!\right]] \\ \qquad \mathsf{handle}(s\sharp r) \\ \qquad (\lambda x : \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_1 \right]\!\right] . \mathbb{T}_e^{\vdash \text{exp}} \left[\!\left[\cdot; \cdot, x : \tau_1; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e' : \tau_2, \theta' \right]\!\right])) \\ \hookrightarrow_\kappa S^*\{(r,l) \mapsto \lambda x : \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_1 \right]\!\right] . \mathbb{T}_e^{\vdash \text{exp}} \left[\!\left[\cdot; \cdot, x : \tau_1; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e' : \tau_2, \theta' \right]\!\right]\}, \langle l \rangle_{s\sharp r}\end{array}}$$

By Lemma 40, we conclude that

$$\cdot, s \mapsto S^*; e^* \hookrightarrow \kappa^{v*\prime}$$
and
$$\cdot, s \mapsto S^*; \kappa^{v*\prime} \hookrightarrow_\kappa S^*\{(r,l) \mapsto \lambda x : \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau_1 \right]\!\right] . \mathbb{T}_e^{\vdash \text{exp}} \left[\!\left[\cdot; \cdot, x : \tau_1; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e' : \tau_2, \theta' \right]\!\right]\}, \langle l \rangle_{s\sharp r}$$

Note that

$$\begin{array}{rcl}
S' & = & S\{(r,l) \mapsto \lambda x : \tau_1.^{\theta'} e\} \\
\mathcal{S}' & = & \mathcal{S}\{(r,l) \mapsto \tau_1 \xrightarrow{\theta'} \tau_2\} \qquad \text{and} \quad \vdash_{\text{stack}} S' : \mathcal{S}' : \overline{\mathcal{S}}' \\
\overline{\mathcal{S}}' & = & \overline{\mathcal{S}}\{(r,l) \mapsto\}
\end{array}$$
and
$$\mathbb{T}_S^{\vdash \text{stack}} \left[\!\left[\vdash_{\text{stack}} S' : \mathcal{S}' : \overline{\mathcal{S}}' \right]\!\right] = S^*\{(r,l) \mapsto \lambda x : \mathbb{T}_\tau^{\vdash \text{type}} [\![\cdot; \Delta \vdash_{\text{type}} \tau_1]\!] . \mathbb{T}_e^{\vdash \text{exp}} \left[\!\left[\cdot; \cdot, x : \tau_1; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e' : \tau_2, \theta' \right]\!\right]\}$$

Finally, note that $\mathbb{T}_v^{\vdash_{\text{cval}}}\left[\!\left[\mathcal{S}' : \overline{\mathcal{S}}' \vdash_{\text{cval}} \langle l\rangle_r : (\mathsf{int}, r)\right]\!\right] = \langle l\rangle_{s\sharp r}$.

Case $\dfrac{\begin{array}{c} S; e_1 \hookrightarrow S_1; v_1 \qquad v_1 \equiv \langle l_1\rangle_{r_1} \\ r_1 \in dom(S_1) \qquad l_1 \in dom(S_1(r_1)) \qquad S_1(r_1, l_1) = \lambda x : \tau_1.^{\theta'} e' \\ S_1; e_2 \hookrightarrow S_2; v_2 \qquad S_2; e'[v_2/x] \hookrightarrow S_3; v_3 \end{array}}{S; e_1\ e_2 \hookrightarrow S_3; v_3}$: ...

Case $\dfrac{\begin{array}{c} S; e_1 \hookrightarrow S_1; v_1 \\ S_1; e_2 \hookrightarrow S_2; v_2 \\ \rho \equiv r \qquad r \in dom(S_2) \qquad l \notin dom(S_2(r)) \end{array}}{S; (e_1, e_2)\ \mathsf{at}\ \rho \hookrightarrow S_2\{(r, l) \mapsto (v_1, v_2)\}; \langle l\rangle_r}$: ...

Case $\dfrac{\begin{array}{c} S; e \hookrightarrow S'; v \qquad v \equiv \langle l\rangle_r \\ r \in dom(S') \qquad l \in dom(S'(r)) \qquad S'(r, l) = (v_1, v_2) \end{array}}{S; \mathsf{fst}\ e \hookrightarrow S'; v_1}$: ...

Case $\dfrac{r \notin dom(S) \qquad S, r \mapsto \{\}; e[r/\varrho] \hookrightarrow S'; v'' \qquad S' \equiv S'', r \mapsto R''}{S; \mathsf{letregion}\ \varrho\ \mathsf{in}\ e \hookrightarrow S''[\bullet/r]; v''[\bullet/r]}$: Note that

$$\dfrac{\begin{array}{c} \cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau \\ \vdash_{\text{ctxt}} \cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}}; r' \\ \cdot, \varrho \succeq \{r'\}; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \varrho \end{array}}{\cdot; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \mathsf{letregion}\ \varrho.e : \tau, r'}$$

and

$$\begin{array}{rcl} e^* & = & \underline{\mathsf{letRGN}}\ [\mathbb{T}_\tau^{\vdash_{\text{place}}}\left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} r'\right]\!\right]] \\ & & [\mathbb{T}_\tau^{\vdash_{\text{type}}}\left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau\right]\!\right]] \\ & & (\Lambda\varrho.\lambda w_\varrho : (\mathbb{T}_\tau^{\vdash_{\text{place}}}\left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{place}} r'\right]\!\right]) \preceq \varrho.\lambda h_\varrho : \mathsf{RGNHandle}\ \varrho. \\ & & \mathbb{T}_e^{\vdash_{\text{exp}}}\left[\!\left[\cdot, \varrho \succeq \{r'\}; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \varrho\right]\!\right]) \\ & = & \underline{\mathsf{letRGN}}\ [s\sharp r']\ [\mathbb{T}_\tau^{\vdash_{\text{type}}}\left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau\right]\!\right]]\ v_\varrho^* \end{array}$$

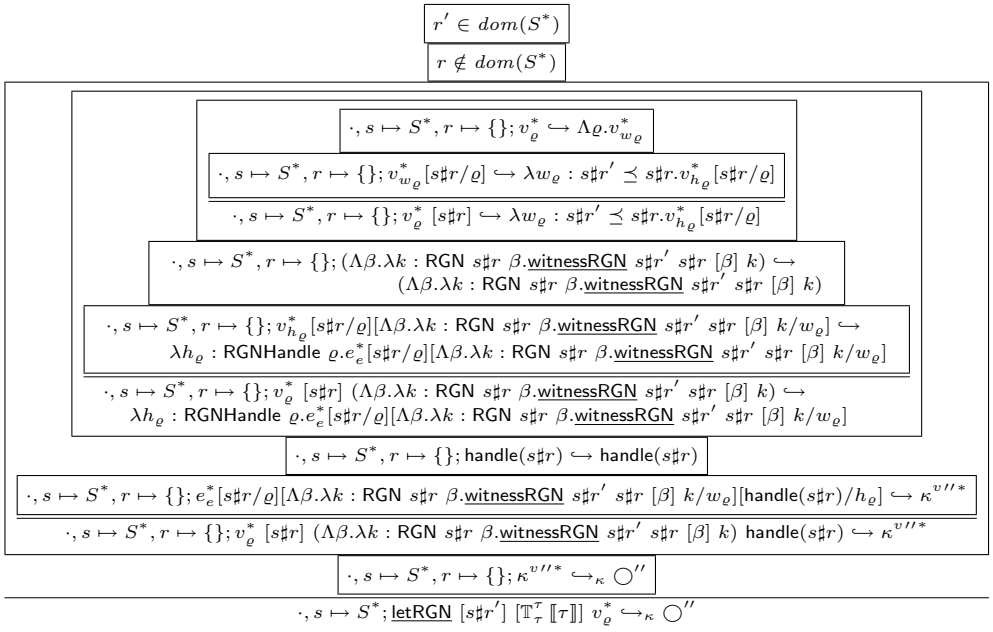
where

$$\begin{array}{rcl} v_\varrho^* & = & \Lambda\varrho.v_{w_\varrho}^* \\ v_{w_\varrho}^* & = & \lambda w_\varrho : (s\sharp r' \preceq \varrho).v_{h_\varrho}^* \\ v_{h_\varrho}^* & = & \lambda h_\varrho : \mathsf{RGNHandle}\ \varrho.e_e^* \\ e_e^* & = & \mathbb{T}_e^{\vdash_{\text{exp}}}\left[\!\left[\cdot, \varrho \succeq \{r'\}; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \varrho\right]\!\right] \end{array}$$

Note that

$$\dfrac{}{\cdot, s \mapsto S^*; e^* \hookrightarrow \underline{\mathsf{letRGN}}\ [s\sharp r']\ [\mathbb{T}_\tau^{\vdash_{\text{type}}}\left[\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\text{type}} \tau\right]\!\right]]\ v_\varrho^*}$$

and



By Lemma 13, we conclude that $\cdot, \varrho \succeq \{r'\}; \cdot; \mathcal{S}, r \mapsto \{\} : \overline{\mathcal{S}}, r \mapsto \{\} \vdash_{\text{exp}} e : \tau, \varrho$. By Lemma 15, we conclude $\tau[r/\varrho] = \tau$.

By Lemma 41, we conclude $e^*_e = \mathbb{T}^{\vdash_{\text{exp}}}_e \left[\!\!\left[\cdot, \varrho \succeq \{r'\}; \cdot; \mathcal{S}, r \mapsto \{\} : \overline{\mathcal{S}}', r \mapsto \{\} \vdash_{\text{exp}} e : \tau, \varrho\right]\!\!\right]$.

By Lemma 42 applied to $\cdot, \varrho \succeq \{r'\}; \cdot; \mathcal{S}, r \mapsto \{\} : \overline{\mathcal{S}}, r \mapsto \{\} \vdash_{\text{exp}} e : \tau, \varrho$ and $\cdot; \overline{\mathcal{S}}, r \mapsto \{\} \vdash_{\text{re}} r \succeq \{r'\}$, we conclude that there exists a derivation of $\cdot; \cdot; \mathcal{S}, r \mapsto \{\} : \overline{\mathcal{S}}, r \mapsto \{\} \vdash_{\text{exp}} e[r/\varrho] : \tau, r$ such that

$$e^*_e[s\sharp r/\varrho][(\Lambda\beta.\lambda k : \mathsf{RGN}\ s\sharp r\ \beta.\underline{\mathsf{witnessRGN}}\ s\sharp r'\ s\sharp r\ [\beta]\ k)/w_\varrho][\mathsf{handle}(s\sharp r)/h_\varrho] = \mathbb{T}^{\vdash_{\text{exp}}}_e \left[\!\!\left[\cdot; \cdot; \mathcal{S}, r \mapsto \{\} : \overline{\mathcal{S}}, r \mapsto \{\} \vdash_{\text{exp}} e[r/\varrho] : \tau, r\right]\!\!\right]$$

Let

$$e'^* = e^*_e[s\sharp r/\varrho][(\Lambda\beta.\lambda k : \mathsf{RGN}\ s\sharp r\ \beta.\underline{\mathsf{witnessRGN}}\ s\sharp r'\ s\sharp r\ [\beta]\ k)/w_\varrho][\mathsf{handle}(s\sharp r)/h_\varrho]$$

By applying the induction hypothesis to $S, r \mapsto \{\}; e[r/\varrho] \hookrightarrow S'', r \mapsto R''; v'', \cdot; \cdot; \mathcal{S}, r \mapsto \{\} : \overline{\mathcal{S}}, r \mapsto \{\} \vdash_{\text{exp}} e[r/\varrho] : \tau, r$, $\mathbb{T}^{\vdash_{\text{exp}}}_e \left[\!\!\left[\cdot; \cdot; \mathcal{S}, r \mapsto \{\} : \overline{\mathcal{S}}, r \mapsto \{\} \vdash_{\text{exp}} e[r/\varrho] : \tau, r\right]\!\!\right] = e'^*$, we conclude that there exists $\overline{\mathcal{S}}'', r \mapsto \overline{\mathcal{R}}'' \supseteq_{=\supseteq} \overline{\mathcal{S}}, r \mapsto \{\}$ and $\mathcal{S}'', r \mapsto \mathcal{R}'' \supseteq_{=\supseteq} \mathcal{S}, r \mapsto \{\}$ such that $\vdash_{\text{stack}} S'', r \mapsto R'' : \mathcal{S}'', r \mapsto \mathcal{R}'' : \overline{\mathcal{S}}'', r \mapsto \overline{\mathcal{R}}''$ and $\mathcal{S}'', r \mapsto \overline{\mathcal{R}}'' : \overline{\mathcal{S}}'', r \mapsto \overline{\mathcal{R}}'' \vdash_{\text{cval}} v'' : \tau$, and $\cdot, s \mapsto S^*, r \mapsto \{\}; e'^* \hookrightarrow \kappa^{v''*}$ and $\cdot, s \mapsto S^*, r \mapsto \{\}; \kappa^{v''*} \hookrightarrow_\kappa S''^*, r \mapsto R''^*; v''^*$ where $\mathbb{T}^{\vdash_{\text{stack}}}_S \left[\!\!\left[\vdash_{\text{stack}} S'', r \mapsto R'' : \mathcal{S}'', r \mapsto \mathcal{R}'' : \overline{\mathcal{S}}'', r \mapsto \overline{\mathcal{R}}''\right]\!\!\right] = S''^*, r \mapsto R''^*$, and $\mathbb{T}^{\vdash_{\text{cval}}}_v \left[\!\!\left[\mathcal{S}'', r \mapsto \mathcal{R}'' : \overline{\mathcal{S}}'', r \mapsto \overline{\mathcal{R}}'' \vdash_{\text{cval}} v'' : \tau\right]\!\!\right] = v''^*$.

Note that

$$\begin{array}{lcl} S' & = & S''[\bullet/r] \\ \mathcal{S}' & = & \mathcal{S}''[\bullet/r] \\ \overline{\mathcal{S}}' & = & \overline{\mathcal{S}}'' \end{array} \quad \text{and} \quad \vdash_{\text{stack}} S' : \mathcal{S}' : \overline{\mathcal{S}}' \quad \text{and} \quad \mathbb{T}^{\vdash_{\text{stack}}}_S \left[\!\!\left[\vdash_{\text{stack}} S' : \mathcal{S}' : \overline{\mathcal{S}}'\right]\!\!\right] = S^*[s\sharp \bullet / s\sharp r]$$

Finally, note that $\mathbb{T}^{\vdash_{\text{cval}}}_v \left[\!\!\left[\mathcal{S}''[\bullet/r] : \overline{\mathcal{S}}''[\bullet/r] \vdash_{\text{cval}} v''[\bullet/r] : \tau\right]\!\!\right] = v''^*[s\sharp \bullet / s\sharp r]$.

Case $\dfrac{\rho \equiv r \qquad r \in dom(S) \qquad l \notin dom(S(r))}{S; \lambda\varrho \succeq \varphi.^{\theta'} u' \ \mathsf{at}\ \rho \hookrightarrow S\{(r,l) \mapsto \lambda\varrho \succeq \varphi.^{\theta'} u'\}; \langle l \rangle_r} : \ldots$

Case $\dfrac{\begin{array}{c} S; e_1 \hookrightarrow S_1; v_1 \qquad v_1 = \langle l_1 \rangle_{r_1} \\ r_1 \in dom(S_1) \qquad l_1 \in dom(S_1(r_1)) \qquad S_1(r_1, l_1) = \lambda \varrho \succeq \varphi.^{\theta'} u' \\ S_1; u'[\rho_2/\varrho] \hookrightarrow S_2; v_2 \end{array}}{S; e_1\ [\rho_2] \hookrightarrow S_2; v_2}$: Note that

$$\dfrac{\begin{array}{c} \cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : (\Pi \varrho \succeq \varphi.^{\theta'} \tau, \rho'), r' \qquad \cdot;\overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq \rho' \\ \cdot;\overline{\mathcal{S}} \vdash_{\text{place}} \rho \qquad \cdot;\overline{\mathcal{S}} \vdash_{\text{re}} \rho \succeq \varphi \\ \cdot;\overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq \theta'[\rho/\varrho] \end{array}}{\cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e\ [\rho] : \tau[\rho/\varrho], r'}$$

Applying Theorem 1 to (1a) $\vdash_{\text{stack}} S : \mathcal{S} : \overline{\mathcal{S}}$, (1b) $\cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : (\Pi \varrho \succeq \varphi.^{\theta'} \tau, \rho'), r'$, and (1c) $S; e \hookrightarrow S'; \langle l \rangle_r$, we conclude that there exists $\overline{\mathcal{S}}' \supseteq_{=\supseteq} \overline{\mathcal{S}}$ and $\mathcal{S}' \supseteq_{=\supseteq} \mathcal{S}$ such that $\vdash_{\text{stack}} S' : \mathcal{S}' : \overline{\mathcal{S}}'$ and $\mathcal{S}' : \overline{\mathcal{S}}' \vdash_{\text{cval}} \langle l \rangle_r : (\Pi \varrho \succeq \varphi.^{\theta'} \tau, \rho')$. Hence, $\rho' = r$.
By applying Lemma 17 to $\cdot;\overline{\mathcal{S}} \vdash_{\text{rr}} r' \succeq \theta'[\rho/\varrho]$, we conclude that $\theta'[\rho/\varrho] = r''$.
By inspection of the derivation $\cdot;\overline{\mathcal{S}} \vdash_{\text{place}} \rho$, we conclude $\rho \equiv r'''$ or $\rho \equiv \bullet$.
Note

$$\begin{array}{rcl} e^* & = & \mathsf{bind}\ f : \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{type}} (\Pi \varrho \succeq \varphi.^{\theta'} \tau, r) \right]\!\right] \Leftarrow \mathbb{T}_e^{\vdash \text{exp}} \left[\!\left[\cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : (\Pi \varrho \succeq \varphi.^{\theta'} \tau, r), r' \right]\!\right]; \\ & & \mathsf{bind}\ g : \mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{type}} \Pi \varrho \succeq \varphi.^{\theta'} \tau \right]\!\right] \\ & & \Leftarrow \mathbb{T}_v^{\vdash \text{rr}} \left[\!\left[\cdot;\mathcal{S} \vdash_{\text{rr}} r' \succeq r \right]\!\right]\ [\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{type}} \Pi \varrho.^{\theta'} \tau \right]\!\right]] \\ & & \qquad (\underline{\mathsf{readRGNVar}}\ [\mathbb{T}_\tau^{\vdash \text{place}} \left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{place}} r \right]\!\right]] \\ & & \qquad\quad [\mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{type}} \Pi \varrho \succeq \varphi.^{\theta'} \tau \right]\!\right]] \\ & & \qquad\quad f); \\ & & \mathsf{let}\ z = (g\ [\mathbb{T}_\tau^{\vdash \text{place}} \left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{place}} \rho \right]\!\right]]\ \mathbb{T}_v^{\vdash \text{re}} \left[\!\left[\cdot;\mathcal{S} \vdash_{\text{re}} \rho \succeq \varphi \right]\!\right]\ \mathbb{T}_e^{\vdash \text{place}} \left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{place}} \rho \right]\!\right])\ \mathsf{in} \\ & & \mathbb{T}_v^{\vdash \text{rr}} \left[\!\left[\cdot;\mathcal{S} \vdash_{\text{rr}} r' \succeq \theta' \right]\!\right]\ \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{type}} \tau[\rho/\varrho] \right]\!\right]\ z \\ & & \quad \text{where } f, g, z \text{ fresh} \\ & = & \mathsf{bind}\ f : \mathsf{RGNVar}\ s\sharp r\ \tau_\Pi \Leftarrow \mathbb{T}_e^{\vdash \text{exp}} \left[\!\left[\cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : (\Pi \varrho.^{\theta'} \tau, r), r' \right]\!\right]; \\ & & \mathsf{bind}\ g : \tau_\Pi \Leftarrow \mathbb{T}_v^{\vdash \text{rr}} \left[\!\left[\cdot;\mathcal{S} \vdash_{\text{rr}} r' \succeq r \right]\!\right]\ [\tau_\Pi]\ (\underline{\mathsf{readRGNVar}}\ [s\sharp r]\ [\tau_\Pi]\ f); \\ & & \mathsf{let}\ z = (g\ [s\sharp\rho]\ \mathbb{T}_v^{\vdash \text{re}} \left[\!\left[\cdot;\mathcal{S} \vdash_{\text{re}} \rho \succeq \varphi \right]\!\right]\ \mathsf{handle}(s\sharp\rho))\ \mathsf{in} \\ & & \mathbb{T}_v^{\vdash \text{rr}} \left[\!\left[\cdot;\mathcal{S} \vdash_{\text{rr}} r' \succeq \theta' \right]\!\right]\ \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{type}} \tau[\rho/\varrho] \right]\!\right]\ z \\ \tau_\Pi & = & \mathbb{T}_\tau^{\vdash \text{btype}} \left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{type}} \Pi \varrho.^{\theta'} \tau \right]\!\right] \\ & = & \forall \varrho.(\mathbb{T}_\tau^{\vdash \text{place}} \left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{place}} \rho_1 \right]\!\right] \preceq \varrho \times \cdots \times \mathbb{T}_\tau^{\vdash \text{place}} \left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{place}} \rho_n \right]\!\right] \preceq \varrho) \rightarrow \\ & & \quad \mathsf{RGNHandle}\ \varrho \rightarrow \\ & & \quad \mathsf{RGN}\ \mathbb{T}_\tau^{\vdash \text{place}} \left[\!\left[\cdot, \varrho \succeq \varphi;\overline{\mathcal{S}} \vdash_{\text{place}} \theta' \right]\!\right]\ \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot, \varrho \succeq \varphi;\overline{\mathcal{S}} \vdash_{\text{type}} \tau \right]\!\right] \\ & & \quad\ \text{where } \varphi = \{\rho_1, \ldots, \rho_n\} \end{array}$$

By applying the induction hypothesis to $S; e \hookrightarrow S'; \langle l \rangle_r$, $\cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : (\Pi \varrho \succeq \varphi.^{\theta'} \tau, r), r'$, $\mathbb{T}_e^{\vdash \text{exp}} \left[\!\left[\cdot;\cdot;\mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : (\Pi \varrho \succeq \varphi.^{\theta'} \tau, r), r' \right]\!\right] = e_1^*$, we conclude that there exists $\overline{\mathcal{S}}' \supseteq_{=\supseteq} \overline{\mathcal{S}}$ and $\mathcal{S}' \supseteq_{=\supseteq} \mathcal{S}$ such that $\vdash_{\text{stack}} S' : \mathcal{S}' : \overline{\mathcal{S}}'$ and $\mathcal{S}' : \overline{\mathcal{S}}' \vdash_{\text{cval}} \langle l \rangle_r : \mathsf{int}$, and $\cdot, s \mapsto S^*; e_1^* \hookrightarrow \kappa_f^{v*}$ and $\cdot, s \mapsto S^*; \kappa_f^{v*} \hookrightarrow_\kappa S'^*; v_f^*$ where $\mathbb{T}_S^{\vdash \text{stack}} \left[\!\left[\vdash_{\text{stack}} S' : \mathcal{S}' : \overline{\mathcal{S}}' \right]\!\right] = S'^*$, and $\mathbb{T}_v^{\vdash \text{cval}} \left[\!\left[\mathcal{S}' : \overline{\mathcal{S}}' \vdash_{\text{cval}} \langle l \rangle_r : (\Pi \varrho \succeq \varphi.^{\theta'} \tau, r) \right]\!\right] = \langle l \rangle_{s\sharp r} = {v_f}^*$.
Note that

$$\begin{array}{rcl} e^* & = & \mathsf{let}\ k_f = e_1^*\ \mathsf{in} \\ & & \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{RGNVar}\ s\sharp r_1\ \tau_\Pi]\ [\mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{type}} \tau[\rho/\varrho] \right]\!\right]]\ k_a\ (\lambda f : \mathsf{RGNVar}\ s\sharp r_1\ \tau_\Pi . e_f^*) \end{array}$$

where

$$\begin{array}{rcl} e_f & = & \mathsf{bind}\ g : \tau_\Pi \Leftarrow \mathbb{T}_v^{\vdash \text{rr}} \left[\!\left[\cdot;\mathcal{S} \vdash_{\text{rr}} r' \succeq r \right]\!\right]\ [\tau_\Pi]\ (\underline{\mathsf{readRGNVar}}\ [s\sharp r]\ [\tau_\Pi]\ f); \\ & & \mathsf{let}\ z = (g\ [s\sharp\rho]\ \mathbb{T}_v^{\vdash \text{re}} \left[\!\left[\cdot;\mathcal{S} \vdash_{\text{re}} \rho \succeq \varphi \right]\!\right]\ \mathsf{handle}(s\sharp\rho))\ \mathsf{in} \\ & & \mathbb{T}_v^{\vdash \text{rr}} \left[\!\left[\cdot;\mathcal{S} \vdash_{\text{rr}} r' \succeq \theta' \right]\!\right]\ \mathbb{T}_\tau^{\vdash \text{type}} \left[\!\left[\cdot;\overline{\mathcal{S}} \vdash_{\text{type}} \tau[\rho/\varrho] \right]\!\right]\ z \end{array}$$

Hence,

$$\frac{\boxed{\cdot, s \mapsto S^*; e_1^* \hookrightarrow \kappa_f^{v\,*}\,*}}{\begin{array}{r}\cdot, s \mapsto S^*; \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{RGNVar}\ s\sharp r\ \tau_\Pi]\ [\mathbb{T}_\tau^{\vdash \mathrm{type}}\ [\![\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{type}} \tau[\rho/\varrho]]\!]]\ \kappa_f^{v\,*}\ (\lambda f : \mathsf{RGNVar}\ s\sharp r\ \tau_\Pi.e_f^*) \\ \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{RGNVar}\ s\sharp r\ \tau_\Pi]\ [\mathbb{T}_\tau^{\vdash \mathrm{type}}\ [\![\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{type}} \tau[\rho/\varrho]]\!]]\ \kappa_f^{v\,*}\ (\lambda f : \mathsf{RGNVar}\ s\sharp r\ \tau_\Pi.e_f^*)\end{array}}$$

Note that

$$\frac{\boxed{\cdot, s \mapsto S^*; \kappa_f^{v\,*} \hookrightarrow_\kappa S'^*; \langle l\rangle_{s\sharp r}} \qquad \frac{\boxed{\cdot, s \mapsto S'^*; e_f^*[\langle l\rangle_{s\sharp r}/f] \hookrightarrow \kappa_F^{v\,*}}}{\cdot, s \mapsto S'^*; (\lambda f : \mathsf{RGNVar}\ s\sharp r\ \tau_\Pi.e_f^*)\ \langle l\rangle_{s\sharp r} \hookrightarrow \kappa_F^{v\,*}} \qquad \boxed{\cdot, s \mapsto S'^*; \kappa_F^{v\,*} \hookrightarrow_\kappa \bigcirc}}{\cdot, s \mapsto S^*; \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\mathsf{RGNVar}\ s\sharp r\ \tau_\Pi]\ [\mathbb{T}_\tau^{\vdash \mathrm{type}}\ [\![\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{type}} \tau[\rho/\varrho]]\!]]\ \kappa_f^{v\,*}\ (\lambda f : \mathsf{RGNVar}\ s\sharp r\ \tau_\Pi.e_f^*) \hookrightarrow_\kappa \bigcirc}$$

Note that

$$\begin{array}{rcl} e_f^*[\langle l\rangle_{s\sharp r}/f] & = & \mathsf{let}\ k_g = \mathbb{T}_v^{\vdash \mathrm{rr}}\ [\![\cdot; \mathcal{S} \vdash_{\mathrm{rr}} r' \succeq r]\!]\ [\tau_\Pi]\ (\underline{\mathsf{readRGNVar}}\ [s\sharp r]\ [\tau_\Pi]\ \langle l\rangle_{s\sharp r})\ \mathsf{in} \\ & & \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\tau_\Pi]\ [\mathbb{T}_\tau^{\vdash \mathrm{type}}\ [\![\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{type}} \tau[\rho/\varrho]]\!]]\ (\lambda g : \tau_\Pi.e_g^*) \end{array}$$

where

$$\begin{array}{rcl} e_g^* & = & \mathsf{let}\ z = (g\ [s\sharp\rho]\ \mathbb{T}_v^{\vdash \mathrm{re}}\ [\![\cdot; \mathcal{S} \vdash_{\mathrm{re}} \rho \succeq \varphi]\!]\ \mathsf{handle}(s\sharp\rho))\ \mathsf{in} \\ & & \mathbb{T}_v^{\vdash \mathrm{rr}}\ [\![\cdot; \mathcal{S} \vdash_{\mathrm{rr}} r' \succeq \theta']\!]\ \mathbb{T}_\tau^{\vdash \mathrm{type}}\ [\![\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{type}} \tau[\rho/\varrho]]\!]\ z \end{array}$$

Note that

$$\frac{\boxed{\cdot; \cdot; \cdot, s \mapsto \mathcal{S}'^* : \cdot, s \mapsto \overline{\mathcal{S}'}^* \vdash_{\mathrm{exp}} \langle l\rangle_{s\sharp r} : \mathsf{RGNVar}\ s\sharp r\ \tau_\Pi}}{\cdot; \cdot; \cdot, s \mapsto \mathcal{S}'^* : \cdot, s \mapsto \overline{\mathcal{S}'}^* \vdash_{\mathrm{exp}} \underline{\mathsf{readRGNVar}}\ [s\sharp r]\ [\tau_\Pi]\ \langle l\rangle_{s\sharp r} : \mathsf{RGN}\ s\sharp r\ \tau_\Pi}$$

and

$$\frac{\boxed{r \in dom(S'^*)} \qquad \boxed{l \in dom(S'^*(r))} \qquad \boxed{v_\varrho = S'^*(r, l)}}{\cdot, s \mapsto S'^*; \underline{\mathsf{readRGNVar}}\ [s\sharp r]\ [\tau_\Pi]\ \langle l\rangle_{s\sharp r} \hookrightarrow_\kappa S'^*; v_\varrho}$$

where

$$\begin{array}{rcl} v_\varrho & = & \mathbb{T}_v^{\vdash \mathrm{sto}}\ \left[\!\!\left[\mathcal{S}'; \overline{\mathcal{S}'} \vdash_{\mathrm{sto}} \lambda\varrho \succeq \varphi.^{\theta'} u' : \Pi\varrho \succeq \varphi.^{\theta'} \tau\right]\!\!\right] \\ & = & \Lambda\varrho.v_{w_\varrho} \\ v_{w_\varrho} & = & \lambda w_\varrho : (\mathbb{T}_\tau^{\vdash \mathrm{place}}\ [\![\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{place}} \rho_1]\!] \preceq \varrho \times \cdots \times \mathbb{T}_\tau^{\vdash \mathrm{place}}\ [\![\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{place}} \rho_n]\!] \preceq \varrho).v_{h_\varrho} \\ v_{h_\varrho} & = & \lambda h_\varrho : \mathsf{RGNHandle}\ \varrho.e_{u'}^* \\ e_{u'}^* & = & \mathbb{T}_e^{\vdash \mathrm{exp}}\ \left[\!\!\left[\cdot, \varrho \succeq \varphi; \cdot; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\mathrm{exp}} u' : \tau, \theta'\right]\!\!\right] \end{array}$$

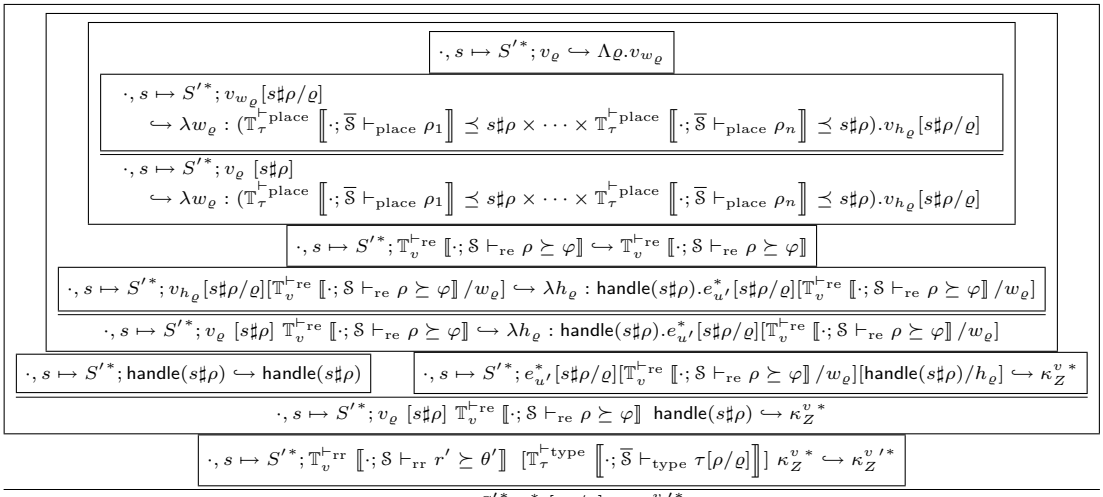
By Lemma 40, we conclude that

$$\begin{array}{c} \cdot, s \mapsto S'^*; \mathbb{T}_v^{\vdash \mathrm{rr}}\ [\![\cdot; \mathcal{S} \vdash_{\mathrm{rr}} r' \succeq r]\!]\ [\tau_\Pi]\ (\underline{\mathsf{readRGNVar}}\ [s\sharp r]\ [\tau_\Pi]\ \langle l\rangle_{s\sharp r}) \hookrightarrow \kappa_g^{v\,*} \\ \text{and} \\ \cdot, s \mapsto S'^*; \kappa_g^{v\,*} \hookrightarrow_\kappa S'^*; v_\varrho \end{array}$$

Note that

$$\frac{\boxed{\cdot, s \mapsto S'^*; \kappa_g^{v\,*} \hookrightarrow_\kappa S'^*; v_\varrho} \qquad \frac{\boxed{\cdot, s \mapsto S'^*; e_g^*\ [v_\varrho/g] \hookrightarrow \kappa_G^{v\,*}}}{\cdot, s \mapsto S'^*; (\lambda g : \tau_\Pi.e_g^*)\ v_\varrho \hookrightarrow \kappa_G^{v\,*}} \qquad \boxed{\cdot, s \mapsto S'^*; \kappa_G^{v\,*} \hookrightarrow_\kappa \bigcirc}}{\cdot, s \mapsto S'^*; \underline{\mathsf{thenRGN}}\ [s\sharp r']\ [\tau_\Pi]\ [\mathbb{T}_\tau^{\vdash \mathrm{type}}\ [\![\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{type}} \tau[\rho/\varrho]]\!]]\ \kappa_g^{v\,*}\ (\lambda g : \tau_\Pi.e_g^*) \hookrightarrow_\kappa \bigcirc}$$

Note that



Note that

$$\frac{\cdot, \varrho \succeq \varphi; \cdot; \mathcal{S}' : \overline{\mathcal{S}'} \vdash_{\mathrm{exp}} u' : \tau, \theta'}{\cdot; \mathcal{S}' : \overline{\mathcal{S}'} \vdash_{\mathrm{sto}} \lambda \varrho \succeq \varphi.^{\theta'} u' : \Pi \varrho \succeq \varphi.^{\theta'} \tau}$$

By Lemma 42 applied to $\cdot, \varrho \succeq \varphi; \cdot; \mathcal{S}' : \overline{\mathcal{S}'} \vdash_{\mathrm{exp}} u' : \tau, \theta'$ and $\cdot; \overline{\mathcal{S}'}^* \vdash_{\mathrm{re}} \rho \succeq \varphi$, we conclude that there exists a derivation of $\cdot; \cdot; \mathcal{S}' : \overline{\mathcal{S}'} \vdash_{\mathrm{exp}} u'[\rho/\varrho] : \tau[\rho/\varrho], \theta'[\rho/\varrho]$ such that

$$\begin{aligned} & e^*_{u'}[s\sharp\rho/\varrho][\mathbb{T}^{\vdash_{\mathrm{re}}}_v [\![\cdot; \mathcal{S} \vdash_{\mathrm{re}} \rho \succeq \varphi]\!] / w_\varrho][\mathsf{handle}(s\sharp\rho)/h_\varrho] \\ & \quad = \mathbb{T}^{\vdash_{\mathrm{exp}}}_e \left[\!\!\left[\cdot; \cdot; \mathcal{S}' : \overline{\mathcal{S}'} \vdash_{\mathrm{exp}} u'[\rho/\varrho] : \tau[\rho/\varrho], \theta'[\rho/\varrho]\right]\!\!\right] \end{aligned}$$

Applying the induction hypothesis to $S'; u'[\rho/\varrho] \hookrightarrow S''; v'', \cdot; \cdot; \mathcal{S}' : \overline{\mathcal{S}'} \vdash_{\mathrm{exp}} u'[\rho/\varrho] : \tau[\rho/\varrho], r''$, $\mathbb{T}^{\vdash_{\mathrm{exp}}}_e [\![\cdot; \cdot; \mathcal{S}' : \overline{\mathcal{S}'} \vdash_{\mathrm{exp}} u'[\rho/\varrho] : \tau[\rho/\varrho], r'']\!] = e^*_{u'}[s\sharp\rho/\varrho][\mathbb{T}^{\vdash_{\mathrm{re}}}_v [\![\cdot; \mathcal{S} \vdash_{\mathrm{re}} \rho \succeq \varphi]\!] / w_\varrho][\mathsf{handle}(s\sharp\rho)/h_\varrho]$, we conclude that there exists $\overline{\mathcal{S}}'' \supseteq_{=\supseteq} \overline{\mathcal{S}}'$ and $\mathcal{S}'' \supseteq_{=\supseteq} \mathcal{S}'$ such that $\vdash_{\mathrm{stack}} S'' : \mathcal{S}'' : \overline{\mathcal{S}}''$ and $\mathcal{S}'' : \overline{\mathcal{S}}'' \vdash_{\mathrm{cval}} v'' : \tau[\rho/\varrho]$, and $\cdot, s \mapsto S'^*; e^*_{u'}[s\sharp\rho/\varrho][\mathbb{T}^{\vdash_{\mathrm{re}}}_v [\![\cdot; \mathcal{S} \vdash_{\mathrm{re}} \rho \succeq \varphi]\!] / w_\varrho][\mathsf{handle}(s\sharp\rho)/h_\varrho] \hookrightarrow \kappa^{v\,*}_Z$ and $\cdot, s \mapsto S'^*; \kappa^{v\,*}_Z \hookrightarrow_\kappa S''^*; v''^*$, where $\mathbb{T}^{\vdash_{\mathrm{stack}}}_S \left[\!\!\left[\vdash_{\mathrm{stack}} S'' : \mathcal{S}'' : \overline{\mathcal{S}}''\right]\!\!\right] = S''^*$, and $\mathbb{T}^{\vdash_{\mathrm{cval}}}_v \left[\!\!\left[\mathcal{S}' : \overline{\mathcal{S}}' \vdash_{\mathrm{cval}} v'' : \tau[\rho/\varrho]\right]\!\!\right] = v''^*$.

Applying Lemma 40, we conclude that $\cdot, s \mapsto S'^*; \mathbb{T}^{\vdash_{\mathrm{rr}}}_v [\![\cdot; \mathcal{S} \vdash_{\mathrm{rr}} r' \succeq \theta']\!]\ \mathbb{T}^{\vdash_{\mathrm{type}}}_\tau \left[\!\!\left[\cdot; \overline{\mathcal{S}} \vdash_{\mathrm{type}} \tau[\rho/\varrho]\right]\!\!\right]\ \kappa^{v\,*}_Z \hookrightarrow \kappa^{v\,\prime *}_Z$ and $\cdot, s \mapsto S'^*; \kappa^{v\,\prime *}_Z \hookrightarrow_\kappa S''^*; v''^*$.

□

Theorem 6 (Correctness)

Suppose $\vdash_{\text{prog}} p\ \mathsf{ok}$ *and* $p \hookrightarrow v'$.
Let $\mathbb{T}_e^{\vdash_{\text{prog}}}[\![\vdash_{\text{prog}} p\ \mathsf{ok}]\!] = e^*$.
Then $\cdot; e^* \hookrightarrow v'^*$, *where* $\mathbb{T}_v^{\vdash_{\text{cval}}}[\![\cdot : \cdot \vdash_{\text{cval}} v' : \mathsf{bool}]\!] = v'^*$.

Proof. Note that

$$\frac{\cdot, \mathsf{H} \mapsto \{\}; p[\mathsf{H}/\mathcal{H}] \hookrightarrow S'; v' \qquad S' \equiv \cdot; \mathsf{H} \mapsto R'}{p \hookrightarrow v'[\bullet/\mathsf{H}]}$$

and

$$\frac{\cdot, \mathcal{H} \succeq \{\}; \cdot; \cdot : \cdot \vdash_{\text{exp}} p : \mathsf{bool}, \mathcal{H}}{\vdash_{\text{prog}} p\ \mathsf{ok}}$$

By Lemma 13, we conclude that $\cdot, \mathcal{H} \succeq \{\}; \cdot; \cdot, \mathsf{H} \mapsto \{\} : \cdot, \mathsf{H} \mapsto \{\} \vdash_{\text{exp}} p : \mathsf{bool}, \mathcal{H}$.

By Lemma 10, we conclude that $\cdot; \cdot; \cdot, \mathsf{H} \mapsto \{\} : \cdot, \mathsf{H} \mapsto \{\} \vdash_{\text{exp}} p[\mathsf{H}/\mathcal{H}] : \mathsf{bool}, \mathsf{H}$.

By Theorem 1, we conclude that there exists $\overline{\mathcal{R}}'$ and $\mathcal{R}'$ such that $\vdash_{\text{stack}} \cdot, \mathsf{H} \mapsto R' : \cdot, \mathsf{H} \mapsto \mathcal{R}' : \cdot, \mathsf{H} \mapsto \overline{\mathcal{R}}'$ and $\cdot, \mathsf{H} \mapsto \mathcal{R}' : \cdot, \mathsf{H} \mapsto \overline{\mathcal{R}}' \vdash_{\text{cval}} v' : \mathsf{bool}$.

Note that

$$\begin{array}{rcl} e^* & = & \mathsf{runRGN}\ [\mathbb{T}_\tau^{\vdash_{\text{type}}}[\![\cdot, \mathcal{H} \succeq \{\}; \cdot \vdash_{\text{type}} \mathsf{bool}]\!]] \\ & & (\Lambda\mathcal{H}.\lambda h_{\mathcal{H}} : \mathsf{RGNHandle}\ \mathcal{H}. \\ & & \ \mathsf{let}\ w_{\mathcal{H}} = ()\ \mathsf{in} \\ & & \ \mathbb{T}_e^{\vdash_{\text{exp}}}[\![\cdot; \mathcal{H} \succeq \{\}; \cdot : \cdot \vdash_{\text{exp}} p : \mathsf{bool}, \mathcal{H}]\!]) \end{array}$$

By Lemma 41, we conclude that

$$\begin{array}{l} \mathbb{T}_e^{\vdash_{\text{exp}}}[\![\cdot; \mathcal{H} \succeq \{\}; \cdot : \cdot \vdash_{\text{exp}} p : \mathsf{bool}, \mathcal{H}]\!] \\ \quad = \mathbb{T}_e^{\vdash_{\text{exp}}}[\![\cdot; \mathcal{H} \succeq \{\}; \cdot, \mathsf{H} \mapsto \{\} : \cdot, \mathsf{H} \mapsto \{\} \vdash_{\text{exp}} p : \mathsf{bool}, \mathcal{H}]\!] \end{array}$$

By Lemma 42, we conclude that

$$\begin{array}{l} \mathbb{T}_e^{\vdash_{\text{exp}}}[\![\cdot; \cdot; \cdot, \mathsf{H} \mapsto \{\} : \cdot, \mathsf{H} \mapsto \{\} \vdash_{\text{exp}} p[\mathsf{H}/\mathcal{H}] : \mathsf{bool}, \mathsf{H}]\!] \\ \quad = \mathbb{T}_e^{\vdash_{\text{exp}}}[\![\cdot; \mathcal{H} \succeq \{\}; \cdot, \mathsf{H} \mapsto \{\} : \cdot, \mathsf{H} \mapsto \{\} \vdash_{\text{exp}} p : \mathsf{bool}, \mathcal{H}]\!]\ [s\sharp\mathsf{H}/\mathcal{H}][()/w_{\mathcal{H}}][\mathsf{handle}(s\sharp\mathsf{H})/h_{\mathcal{H}}] \\ \quad = \mathbb{T}_e^{\vdash_{\text{exp}}}[\![\cdot; \mathcal{H} \succeq \{\}; \cdot : \cdot \vdash_{\text{exp}} p : \mathsf{bool}, \mathcal{H}]\!]\ [s\sharp\mathsf{H}/\mathcal{H}][()/w_{\mathcal{H}}][\mathsf{handle}(s\sharp\mathsf{H})/h_{\mathcal{H}}] \end{array}$$

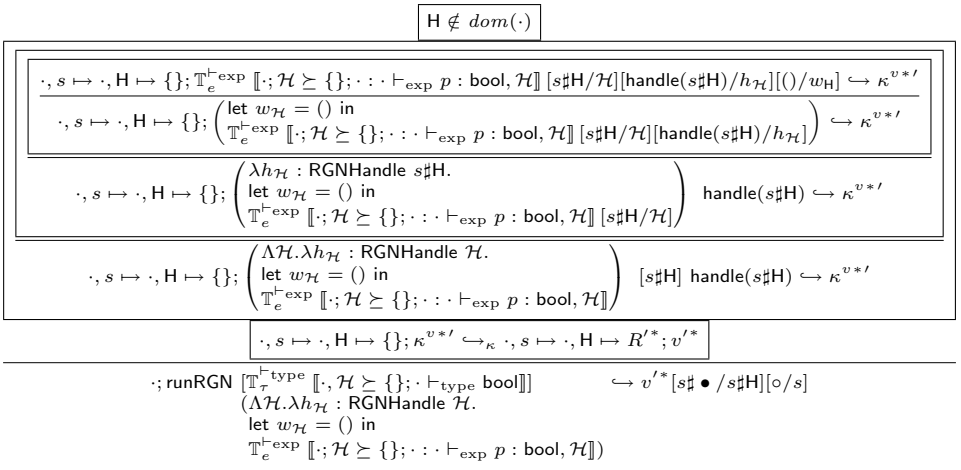
Applying Theorem 5 to $\vdash_{\text{stack}} \cdot, \mathsf{H} \mapsto \{\} : \cdot, \mathsf{H} \mapsto \{\} : \cdot, \mathsf{H} \mapsto \{\}$, $\cdot; \cdot; \cdot, \mathsf{H} \mapsto \{\} : \cdot, \mathsf{H} \mapsto \{\} \vdash_{\text{exp}} p[\mathsf{H}/\mathcal{H}] : \mathsf{bool}, \mathsf{H}$, and $\cdot, \mathsf{H} \mapsto \{\}; p[\mathsf{H}/\mathcal{H}] \hookrightarrow \cdot, \mathsf{H} \mapsto R'; v'$, we conclude that

$$\begin{array}{c} \cdot, s \mapsto \cdot, \mathsf{H} \mapsto \{\}; \mathbb{T}_e^{\vdash_{\text{exp}}}[\![\cdot; \cdot; \cdot, \mathsf{H} \mapsto \{\} : \cdot, \mathsf{H} \mapsto \{\} \vdash_{\text{exp}} p[\mathsf{H}/\mathcal{H}] : \mathsf{bool}, \mathsf{H}]\!] \hookrightarrow \kappa^{v*\prime} \\ \text{and} \\ \cdot, s \mapsto \cdot, \mathsf{H} \mapsto \{\}; \kappa^{v*\prime} \hookrightarrow_\kappa \cdot, s \mapsto \cdot, \mathsf{H} \mapsto R'^*; v'^* \end{array}$$

where

$$\begin{array}{c} \mathbb{T}_S^{\vdash_{\text{stack}}}\left[\!\!\left[\vdash_{\text{stack}} \cdot, \mathsf{H} \mapsto R' : \cdot, \mathsf{H} \mapsto \mathcal{R}' : \cdot, \mathsf{H} \mapsto \overline{\mathcal{R}'}\right]\!\!\right] = s \mapsto \cdot, \mathsf{H} \mapsto R'^* \\ \text{and} \\ \mathbb{T}_v^{\vdash_{\text{cval}}}\left[\!\!\left[\cdot, \mathsf{H} \mapsto \mathcal{R}' : \cdot, \mathsf{H} \mapsto \overline{\mathcal{R}'} \vdash_{\text{cval}} v' : \mathsf{bool}\right]\!\!\right] = v'^* \end{array}$$

Hence,



Recall that $\cdot, \mathsf{H} \mapsto \mathcal{R}' : \cdot, \mathsf{H} \mapsto \overline{\mathcal{R}}' \vdash_{\text{cval}} v' : \mathsf{bool}$. Proceed by inpsection of the derivation.

Case $\dfrac{\vdash_{\text{stype}} \cdot, \mathsf{H} \mapsto \mathcal{R}' : \cdot, \mathsf{H} \mapsto \overline{\mathcal{R}}'}{\cdot, \mathsf{H} \mapsto \mathcal{R}' : \cdot, \mathsf{H} \mapsto \overline{\mathcal{R}}' \vdash_{\text{cval}} \mathsf{tt} : \mathsf{bool}}$: Note that

$$\mathbb{T}_v^{\vdash \text{cval}} \left[\!\!\left[\cdot, \mathsf{H} \mapsto \mathcal{R}' : \cdot, \mathsf{H} \mapsto \overline{\mathcal{R}}' \vdash_{\text{cval}} \mathsf{tt} : \mathsf{bool}\right]\!\!\right] = \mathsf{tt}$$
$$\text{and}$$
$$\mathsf{tt}[s\sharp \bullet /s\sharp\mathsf{H}][\circ/s] = \mathsf{tt}$$
$$\text{and}$$
$$\mathbb{T}_v^{\vdash \text{cval}} [\![\cdot : \cdot \vdash_{\text{cval}} \mathsf{tt}[\bullet/\mathsf{H}] : \mathsf{bool}]\!] = \mathsf{tt}$$

Case $\dfrac{\vdash_{\text{stype}} \cdot, \mathsf{H} \mapsto \mathcal{R}' : \cdot, \mathsf{H} \mapsto \overline{\mathcal{R}}'}{\cdot, \mathsf{H} \mapsto \mathcal{R}' : \cdot, \mathsf{H} \mapsto \overline{\mathcal{R}}' \vdash_{\text{cval}} \mathsf{ff} : \mathsf{bool}}$: Note that

$$\mathbb{T}_v^{\vdash \text{cval}} \left[\!\!\left[\cdot, \mathsf{H} \mapsto \mathcal{R}' : \cdot, \mathsf{H} \mapsto \overline{\mathcal{R}}' \vdash_{\text{cval}} \mathsf{ff} : \mathsf{bool}\right]\!\!\right] = \mathsf{ff}$$
$$\text{and}$$
$$\mathsf{ff}[s\sharp \bullet /s\sharp\mathsf{H}][\circ/s] = \mathsf{ff}$$
$$\text{and}$$
$$\mathbb{T}_v^{\vdash \text{cval}} [\![\cdot : \cdot \vdash_{\text{cval}} \mathsf{ff}[\bullet/\mathsf{H}] : \mathsf{bool}]\!] = \mathsf{ff}$$

□

$$
\begin{array}{rrcl}
 & i & \in & \mathbb{Z} \\
 & \varepsilon & \in & EVars^{TT} \\
 & \vartheta, \varrho & \in & RVars^{TT} \qquad \text{where } \mathcal{H} \in RVars^{TT} \\
 & f, x & \in & Vars^{TT} \\
 & & & \\
\text{Surface region placeholders} & \theta, \rho & ::= & \varrho \\
\text{Surface effects} & \varphi & ::= & \emptyset \mid \{\rho\} \mid \varepsilon \mid \varphi_1 \cup \varphi_2 \\
\text{Surface types} & \tau & ::= & \mathsf{bool} \mid (\mu, \rho) \mid \forall \varepsilon.\tau \\
\text{Surface boxed types} & \mu & ::= & \mathsf{int} \mid \tau_1 \xrightarrow{\varphi} \tau_2 \mid \tau_1 \times \tau_2 \mid \Pi \varrho.^{\varphi} \tau \\
 & & & \\
\text{Surface programs} & p & ::= & e \\
\text{Surface terms} & e & ::= & i \ \mathsf{at}\ \rho \mid e_1 \oplus e_2 \ \mathsf{at}\ \rho \mid e_1 \otimes e_2 \mid \\
 & & & \mathsf{tt} \mid \mathsf{ff} \mid \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f \mid \\
 & & & x \mid \lambda x : \tau.^{\varphi} e \ \mathsf{at}\ \rho \mid e_1\ e_2 \mid \\
 & & & (e_1, e_2)\ \mathsf{at}\ \rho \mid \mathsf{fst}\ e \mid \mathsf{snd}\ e \mid \\
 & & & \mathsf{letregion}\ \varrho.e \mid \lambda \varrho.^{\varphi} u \ \mathsf{at}\ \rho \mid e\ [\rho] \mid \\
 & & & \mathsf{fix}\ f : \tau.u \\
\text{Surface abstractions} & u & ::= & \lambda x : \tau.^{\varphi} e \ \mathsf{at}\ \rho \mid \lambda \varrho.^{\varphi} u \ \mathsf{at}\ \rho \\
\text{Surface values} & v & ::= & \mathsf{tt} \mid \mathsf{ff} \mid x
\end{array}
$$

Figure 51: Surface syntax of TT

5 Expressiveness of the Single Effect Calculus

An important issue to consider is the expressiveness of the Single Effect Calculus relative to Tofte and Talpin's original region calculus. Tofte and Talpin's formulation of the region calculus as the implicit target of an inference system makes a direct comparison difficult. Fortunately, there has been sufficient interest in region-based memory management to warrant direct presentations of region calculi [10, 3, 4, 11], which are better suited for comparison. Three aspects of the region calculus are highlighted as essential features: region polymorphism, region polymorphic recursion, and effect polymorphism.

5.1 Tofte-Talpin

For comparision, we adopt the presentation of the Tofte-Talpin region calculus given in by Henglein, Makholm, and Niss [11]. We have added integer and product types and made the effect/region context explicit. We have dropped type polymorphism, as it does not appear in the Single Effect Calculus (or the Bounded Region Calculus); adding it to both calculi is trivial and orthogonal to the comparisons we wish to make. Finally, we give a distinguished rule for top-level surface programs, to establish "boundary conditions" for a program's execution.

5.1.1 Surface Syntax of TT

Figure 51 presents the syntax of "surface programs" (that is, excluding intermediate terms that appear in the operational semantics) of the Tofte-Talpin Calculus.

As in the Single Effect Calculus, we distinguish between places and effects. The Tofte-Talpin Calculus includes a richer class of effects – essentially, that of a finite set of places and effect variables. We also distinguish between types and boxed types; note that the "single effect" θ in function and region abstraction boxed types of the Single Effect Calculus are replaced by a "general effect" φ in the Tofte-Talpin Calculus. Also note that region abstraction types do not include a region bound. Finally, the Tofte-Talpin admits universal quantification of effects over types: $\forall \varepsilon.\tau$.

Effect/Region contexts	Δ	::=	$\cdot \mid \Delta, \varepsilon \mid \Delta, \varrho$
Value contexts	Γ	::=	$\cdot \mid \Gamma, x : \tau$

Figure 52: Static semantics of TT (definitions)

The Tofte-Talpin Calculus includes all the terms seen previously in the Single Effect Calculus. Once again, note that the "single effect" θ in function and region abstractions are replaced by a general effect φ in the Tofte-Talpin Calculus. Also note that region abstractions do not include a region bound. Finally, the Tofte-Talpin include a fixed-point term, $\mathsf{fix}\ f : \tau.u$. Since we intend the Tofte-Talpin Calculus to obey a call-by-value evaluation semantics, we limit the body of a fixed-point to abstractions.

5.1.2 Static Semantics of TT

Figures 52, 53, 54, 55, 56, and 57 define the static semantics of the Tofte-Talpin Calculus. The development is entirely analagous to that of the Single Effect Calculus.

We summarize the main typing judgements in the following table:

Judgement	Meaning
$\Delta \vdash_{\text{btype}} \mu$	Boxed type μ is well-formed.
$\Delta \vdash_{\text{type}} \tau$	Type τ is well-formed.
$\Delta \vdash_{\text{er}} \varphi \ni \rho$	Region ρ is a region in φ.
$\Delta \vdash_{\text{ee}} \varphi \supseteq \varphi'$	All region in φ' are regions in φ.
$\Delta; \Gamma \vdash_{\text{exp}} e : \tau, \varphi$	Term e has type τ and effect φ.
$\vdash_{\text{prog}} p\ \mathsf{ok}$	Program p is well-typed.

5.2 Region Polymorphism

The Single Effect Calculus clearly supports region polymorphism, albeit, in a slightly different form than that usually found in region calculi. One can give a straightforward translation from the Tofte-Talpin region region calculus without effect polymorphism (or fix) into the Single Effect Calculus. This translation is essentially the same as that of Section 2.8.3; we include it here for completeness.

We start with a few preliminaries. We assume injections from the sets $RVars^{TT}$ and $Vars^{TT}$ to the sets $RVars^{SEC}$ and $Vars^{SEC}$ respectively. In the translation, applications of such injections will be clear from context and we freely use variables from source objects in target objects. We further assume a partitioning

$$RVars^{SEC} = RVars^{TT} \uplus \Theta$$

and draw ϑ region variables from the set Θ. Hence, no ϑ region variable appears in any source TT program.

Note that in the absence of effect polymorphism, effects in the Tofte-Talpin calculus are of the form $\varphi ::= \emptyset \mid \{\rho\} \mid \varphi_1 \cup \varphi_2$. Hence, it is clear that effects can be given in the form $\varphi ::= \{\rho_1, \ldots, \rho_n\}$. In light of the injection from $RVars^{TT}$ to $RVars^{SEC}$, we can freely use effects from source objects in target objects.

Figure 58 gives the translation from Tofte-Talpin Calculus to the Single Effect Calculus. The proof that the translation is type-preserving is virtually identical to that of Section 2.8.3.

Lemma 44 (Translation preserves types)

(1) If $\vdash_{\text{erctxt}} \Delta$, *then* $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!]$.
Furthermore, $dom(\Delta) = dom(\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!])$.

(2) If $\Delta \vdash_{\text{place}} \rho$, *then* $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!] \vdash_{\text{place}} \rho$.

(3) If $\Delta \vdash_{\text{eff}} \varphi$, *then* $\mathbb{T}^{\Delta}_{\Delta}[\![\Delta]\!] \vdash_{\text{eff}} \varphi$.

$\boxed{\Delta; \Gamma \vdash_{\text{exp}} e : \tau, \varphi}$

$$\frac{\vdash_{\text{ctxt}} \Delta; \Gamma; \varphi \qquad \Delta \vdash_{\text{place}} \rho \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho}{\Delta; \Gamma \vdash_{\text{exp}} i \text{ at } \rho : (\mathsf{int}, \rho), \varphi}$$

$$\frac{\Delta; \Gamma \vdash_{\text{exp}} e_1 : (\mathsf{int}, \rho_1), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho_1 \qquad \Delta; \Gamma \vdash_{\text{exp}} e_2 : (\mathsf{int}, \rho_2), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho_2 \qquad \Delta \vdash_{\text{place}} \rho \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho}{\Delta; \Gamma \vdash_{\text{exp}} e_1 \oplus e_2 \text{ at } \rho : (\mathsf{int}, \rho), \varphi}$$

$$\frac{\Delta; \Gamma \vdash_{\text{exp}} e_1 : (\mathsf{int}, \rho_1), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho_1 \qquad \Delta; \Gamma \vdash_{\text{exp}} e_2 : (\mathsf{int}, \rho_2), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho_2}{\Delta; \Gamma \vdash_{\text{exp}} e_1 \otimes e_2 : \mathsf{bool}, \varphi}$$

$$\frac{\vdash_{\text{ctxt}} \Delta; \Gamma; \varphi}{\Delta; \Gamma \vdash_{\text{exp}} \mathsf{tt} : \mathsf{bool}, \varphi} \qquad \frac{\vdash_{\text{ctxt}} \Delta; \Gamma; \varphi}{\Delta; \Gamma \vdash_{\text{exp}} \mathsf{ff} : \mathsf{bool}, \varphi}$$

$$\frac{\Delta; \Gamma \vdash_{\text{exp}} e_b : \mathsf{bool}, \varphi \qquad \Delta; \Gamma \vdash_{\text{exp}} e_t : \tau, \varphi \qquad \Delta; \Gamma \vdash_{\text{exp}} e_f : \tau, \varphi}{\Delta; \Gamma \vdash_{\text{exp}} \mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f : \tau, \varphi}$$

$$\frac{\vdash_{\text{ctxt}} \Delta; \Gamma; \varphi \qquad x \in dom(\Gamma) \qquad \Gamma(x) = \tau}{\Delta; \Gamma \vdash_{\text{exp}} x : \tau, \varphi}$$

$$\frac{\Delta; \Gamma, x : \tau_1 \vdash_{\text{exp}} e' : \tau_2, \varphi' \qquad \Delta \vdash_{\text{place}} \rho \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho}{\Delta; \Gamma \vdash_{\text{exp}} \lambda x : \tau_1.^{\varphi'} e' \text{ at } \rho : (\tau_1 \xrightarrow{\varphi'} \tau_2, \rho), \varphi}$$

$$\frac{\Delta; \Gamma \vdash_{\text{exp}} e_1 : (\tau_1 \xrightarrow{\varphi'} \tau_2, \rho_1'), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho_1' \qquad \Delta; \Gamma \vdash_{\text{exp}} e_2 : \tau_1, \varphi \qquad \Delta \vdash_{\text{ee}} \varphi \supseteq \varphi'}{\Delta; \Gamma \vdash_{\text{exp}} e_1\ e_2 : \tau_2, \varphi}$$

$$\frac{\Delta; \Gamma \vdash_{\text{exp}} e_1 : \tau_1, \varphi \qquad \Delta; \Gamma \vdash_{\text{exp}} e_2 : \tau_2, \varphi \qquad \Delta \vdash_{\text{place}} \rho \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho}{\Delta; \Gamma \vdash_{\text{exp}} (e_1, e_2) \text{ at } \rho : (\tau_1 \times \tau_2, \rho), \varphi}$$

$$\frac{\Delta; \Gamma \vdash_{\text{exp}} e : (\tau_1 \times \tau_2, \rho), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho}{\Delta; \Gamma \vdash_{\text{exp}} \mathsf{fst}\ \ e : \tau_1, \varphi}$$

$$\frac{\Delta; \Gamma \vdash_{\text{exp}} e : (\tau_1 \times \tau_2, \rho), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho}{\Delta; \Gamma \vdash_{\text{exp}} \mathsf{snd}\ \ e : \tau_2, \varphi}$$

$$\frac{\Delta \vdash_{\text{type}} \tau \qquad \vdash_{\text{ctxt}} \Delta; \Gamma; \varphi \qquad \Delta, \varrho; \Gamma \vdash_{\text{exp}} e : \tau, \varphi \cup \{\varrho\}}{\Delta; \Gamma \vdash_{\text{exp}} \mathsf{letregion}\ \varrho.e : \tau, \varphi}$$

$$\frac{\Delta, \varrho; \Gamma \vdash_{\text{exp}} u' : \tau, \varphi' \qquad \Delta \vdash_{\text{place}} \rho \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho}{\Delta; \Gamma \vdash_{\text{exp}} \lambda \varrho.^{\varphi'} u' \text{ at } \rho : (\Pi \varrho.^{\varphi'} \tau, \rho), \varphi}$$

$$\frac{\Delta; \Gamma \vdash_{\text{exp}} e : (\Pi \varrho.^{\varphi'} \tau, \rho_1'), \varphi \qquad \Delta \vdash_{\text{er}} \varphi \ni \rho_1' \qquad \Delta \vdash_{\text{place}} \rho_2 \qquad \Delta \vdash_{\text{ee}} \varphi \supseteq \varphi'[\rho_2/\varrho]}{\Delta; \Gamma \vdash_{\text{exp}} e\ [\rho_2] : \tau[\rho_2/\varrho], \varphi}$$

$$\frac{\Delta; \Gamma, f : \tau \vdash_{\text{exp}} u : \tau, \varphi}{\Delta; \Gamma \vdash_{\text{exp}} \mathsf{fix}\ f : \tau.u : \tau, \varphi}$$

$$\frac{\vdash_{\text{ctxt}} \Delta; \Gamma; \varphi \qquad \Delta, \varepsilon; \Gamma \vdash_{\text{exp}} e : \tau, \varphi}{\Delta; \Gamma \vdash_{\text{exp}} e : \forall \varepsilon. \tau, \varphi}$$

$$\frac{\Delta; \Gamma \vdash_{\text{exp}} e : \forall \varepsilon. \tau, \varphi \qquad \Delta \vdash_{\text{eff}} \varphi'}{\Delta; \Gamma \vdash_{\text{exp}} e : \tau[\varphi'/\varepsilon], \varphi}$$

Figure 53: Static semantics of TT (expressions)

$\boxed{\Delta \vdash_{\text{er}} \varphi \ni \rho}$

$$\frac{\Delta \vdash_{\text{place}} \rho}{\Delta \vdash_{\text{er}} \{\rho\} \ni \rho} \qquad \frac{\Delta \vdash_{\text{eff}} \varphi_2 \qquad \Delta \vdash_{\text{er}} \varphi_1 \ni \rho}{\Delta \vdash_{\text{er}} \varphi_1 \cup \varphi_2 \ni \rho} \qquad \frac{\Delta \vdash_{\text{eff}} \varphi_1 \qquad \Delta \vdash_{\text{er}} \varphi_2 \ni \rho}{\Delta \vdash_{\text{er}} \varphi_1 \cup \varphi_2 \ni \rho}$$

$\boxed{\Delta \vdash_{\text{re}} \varphi \supseteq \varphi'}$

$$\frac{\Delta \vdash_{\text{eff}} \varphi}{\Delta \vdash_{\text{ee}} \varphi \supseteq \emptyset} \qquad \frac{\Delta \vdash_{\text{er}} \varphi \ni \rho}{\Delta \vdash_{\text{ee}} \varphi \supseteq \{\rho\}} \qquad \frac{\Delta \vdash_{\text{ee}} \varphi \supseteq \varphi_1 \qquad \Delta \vdash_{\text{ee}} \varphi \supseteq \varphi_2}{\Delta \vdash_{\text{ee}} \varphi \supseteq \varphi_1 \cup \varphi_2} \qquad \frac{\vdash_{\text{erctxt}} \Delta \qquad \varepsilon \in dom(\Delta)}{\Delta \vdash_{\text{er}} \varepsilon \ni \varepsilon}$$

$$\frac{\Delta \vdash_{\text{eff}} \varphi_2 \qquad \Delta \vdash_{\text{ee}} \varphi_1 \supseteq \varepsilon}{\Delta \vdash_{\text{ee}} \varphi_1 \cup \varphi_2 \supseteq \varepsilon} \qquad \frac{\Delta \vdash_{\text{eff}} \varphi_1 \qquad \Delta \vdash_{\text{ee}} \varphi_2 \supseteq \varepsilon}{\Delta \vdash_{\text{ee}} \varphi_1 \cup \varphi_2 \supseteq \varepsilon}$$

Figure 54: Static semantics of TT (casts)

$\boxed{\Delta \vdash_{\text{place}} \rho}$

$$\frac{\vdash_{\text{erctxt}} \Delta \qquad \varrho \in dom(\Delta)}{\Delta \vdash_{\text{place}} \varrho}$$

$\boxed{\Delta \vdash_{\text{eff}} \varphi}$

$$\frac{\vdash_{\text{erctxt}} \Delta}{\Delta \vdash_{\text{eff}} \emptyset} \qquad \frac{\Delta \vdash_{\text{place}} \rho}{\Delta \vdash_{\text{eff}} \{\rho\}} \qquad \frac{\vdash_{\text{ctxt}} \Delta \qquad \varepsilon \in dom(\Delta)}{\Delta \vdash_{\text{eff}} \varepsilon} \qquad \frac{\Delta \vdash_{\text{eff}} \varphi_1 \qquad \Delta \vdash_{\text{eff}} \varphi_2}{\Delta \vdash_{\text{eff}} \varphi_1 \cup \varphi_2}$$

$\boxed{\Delta \vdash_{\text{btype}} \mu}$

$$\frac{\vdash_{\text{erctxt}} \Delta}{\Delta \vdash_{\text{btype}} \mathsf{int}} \qquad \frac{\Delta \vdash_{\text{type}} \tau_1 \qquad \Delta \vdash_{\text{eff}} \varphi \qquad \Delta \vdash_{\text{type}} \tau_2}{\Delta \vdash_{\text{btype}} \tau_1 \xrightarrow{\varphi} \tau_2} \qquad \frac{\Delta \vdash_{\text{type}} \tau_1 \qquad \Delta \vdash_{\text{type}} \tau_2}{\Delta \vdash_{\text{btype}} \tau_1 \times \tau_2}$$

$$\frac{\Delta, \varrho \vdash_{\text{eff}} \varphi \qquad \Delta, \varrho \vdash_{\text{type}} \tau}{\Delta \vdash_{\text{btype}} \Pi \varrho.^{\varphi} \tau}$$

$\boxed{\Delta \vdash_{\text{type}} \tau}$

$$\frac{\vdash_{\text{erctxt}} \Delta}{\Delta \vdash_{\text{type}} \mathsf{bool}} \qquad \frac{\Delta \vdash_{\text{btype}} \mu \qquad \Delta \vdash_{\text{place}} \rho}{\Delta \vdash_{\text{type}} (\mu, \rho)} \qquad \frac{\Delta, \varepsilon \vdash_{\text{type}} \tau}{\Delta \vdash_{\text{type}} \forall \varepsilon.\tau}$$

Figure 55: Static semantics of TT (types)

$\boxed{\vdash_{\text{erctxt}} \Delta}$

$$\frac{}{\vdash_{\text{erctxt}} \cdot} \qquad \frac{\vdash_{\text{erctxt}} \Delta \qquad \varrho \notin dom(\Delta)}{\vdash_{\text{erctxt}} \Delta, \varrho} \qquad \frac{\vdash_{\text{erctxt}} \Delta \qquad \varepsilon \notin dom(\Delta)}{\vdash_{\text{erctxt}} \Delta, \varepsilon}$$

$\boxed{\Delta \vdash_{\text{vctxt}} \Gamma}$

$$\frac{\vdash_{\text{erctxt}} \Delta}{\Delta \vdash_{\text{vctxt}} \cdot} \qquad \frac{\Delta \vdash_{\text{vctxt}} \Gamma \qquad x \notin dom(\Gamma) \qquad \Delta \vdash_{\text{type}} \tau}{\Delta \vdash_{\text{vctxt}} \Gamma, x : \tau}$$

$\boxed{\vdash_{\text{ctxt}} \Delta; \Gamma; \varphi}$

$$\frac{\Delta \vdash_{\text{vctxt}} \Gamma \qquad \Delta \vdash_{\text{eff}} \varphi}{\vdash_{\text{ctxt}} \Delta; \Gamma; \varphi}$$

Figure 56: Static semantics of TT (contexts)

$\vdash_{\text{prog}} p\ \mathsf{ok}$

$$\frac{\cdot, \mathcal{H}; \cdot \vdash_{\text{exp}} p : \mathsf{bool}, \{\mathcal{H}\}}{\vdash_{\text{prog}} p\ \mathsf{ok}}$$

Figure 57: Static semantics of TT (surface programs)

Region contexts

$$\begin{array}{rcl}
\mathbb{T}^{\Delta}_{\Delta}\left[\!\left[\cdot\right]\!\right] & = & \cdot \\
\mathbb{T}^{\Delta}_{\Delta}\left[\!\left[\Delta, \varrho\right]\!\right] & = & \mathbb{T}^{\Delta}_{\Delta}\left[\!\left[\Delta\right]\!\right], \varrho \succeq \{\}
\end{array}$$

Boxed types

$$\begin{array}{rcl}
\mathbb{T}^{\mu}_{\mu}\left[\!\left[\mathsf{int}\right]\!\right]_{\rho} & = & \mathsf{int} \\
\mathbb{T}^{\mu}_{\mu}\left[\!\left[\tau_1 \xrightarrow{\varphi} \tau_2\right]\!\right]_{\rho} & = & \Pi\vartheta \succeq \varphi.^{\rho}(\mathbb{T}^{\tau}_{\tau}\left[\!\left[\tau_1\right]\!\right] \xrightarrow{\vartheta} \mathbb{T}^{\tau}_{\tau}\left[\!\left[\tau_2\right]\!\right], \rho) \\
\mathbb{T}^{\mu}_{\mu}\left[\!\left[\tau_1 \times \tau_2\right]\!\right]_{\rho} & = & \mathbb{T}^{\tau}_{\tau}\left[\!\left[\tau_1\right]\!\right] \times \mathbb{T}^{\tau}_{\tau}\left[\!\left[\tau_2\right]\!\right] \\
\mathbb{T}^{\mu}_{\mu}\left[\!\left[\Pi\varrho.^{\varphi}\tau\right]\!\right]_{\rho} & = & \Pi\varrho \succeq \{\}.^{\rho}(\Pi\vartheta \succeq \varphi.^{\vartheta}\mathbb{T}^{\tau}_{\tau}\left[\!\left[\tau\right]\!\right], \rho)
\end{array}$$

Types

$$\begin{array}{rcl}
\mathbb{T}^{\tau}_{\tau}\left[\!\left[\mathsf{bool}\right]\!\right] & = & \mathsf{bool} \\
\mathbb{T}^{\tau}_{\tau}\left[\!\left[(\mu, \rho)\right]\!\right] & = & \mathbb{T}^{\mu}_{\mu}\left[\!\left[\mu\right]\!\right]_{\rho}
\end{array}$$

Value contexts

$$\begin{array}{rcl}
\mathbb{T}^{\Gamma}_{\Gamma}\left[\!\left[\cdot\right]\!\right] & = & \cdot \\
\mathbb{T}^{\Gamma}_{\Gamma}\left[\!\left[\Gamma, x : \tau\right]\!\right] & = & \mathbb{T}^{\Gamma}_{\Gamma}\left[\!\left[\Gamma\right]\!\right], x : \mathbb{T}^{\tau}_{\tau}\left[\!\left[\tau\right]\!\right]
\end{array}$$

Expressions

$$\begin{array}{rcl}
\mathbb{T}^{e}_{e}\left[\!\left[i\ \mathsf{at}\ \rho\right]\!\right]_{\theta} & = & i\ \mathsf{at}\ \rho \\
\mathbb{T}^{e}_{e}\left[\!\left[e_1 \oplus e_2\ \mathsf{at}\ \rho\right]\!\right]_{\theta} & = & \mathbb{T}^{e}_{e}\left[\!\left[e_1\right]\!\right]_{\theta} \oplus \mathbb{T}^{e}_{e}\left[\!\left[e_2\right]\!\right]_{\theta}\ \mathsf{at}\ \rho \\
\mathbb{T}^{e}_{e}\left[\!\left[e_1 \otimes e_2\right]\!\right]_{\theta} & = & \mathbb{T}^{e}_{e}\left[\!\left[e_1\right]\!\right]_{\theta} \otimes \mathbb{T}^{e}_{e}\left[\!\left[e_2\right]\!\right]_{\theta} \\
\mathbb{T}^{e}_{e}\left[\!\left[\mathsf{tt}\right]\!\right]_{\theta} & = & \mathsf{tt} \\
\mathbb{T}^{e}_{e}\left[\!\left[\mathsf{ff}\right]\!\right]_{\theta} & = & \mathsf{ff} \\
\mathbb{T}^{e}_{e}\left[\!\left[\mathsf{if}\ e_b\ \mathsf{then}\ e_t\ \mathsf{else}\ e_f\right]\!\right]_{\theta} & = & \mathsf{if}\ \mathbb{T}^{e}_{e}\left[\!\left[e_b\right]\!\right]_{\theta}\ \mathsf{then}\ \mathbb{T}^{e}_{e}\left[\!\left[e_t\right]\!\right]_{\theta}\ \mathsf{else}\ \mathbb{T}^{e}_{e}\left[\!\left[e_f\right]\!\right]_{\theta} \\
\mathbb{T}^{e}_{e}\left[\!\left[x\right]\!\right]_{\theta} & = & x \\
\mathbb{T}^{e}_{e}\left[\!\left[\lambda x : \tau.^{\varphi} e\ \mathsf{at}\ \rho\right]\!\right]_{\theta} & = & \lambda\vartheta \succeq \varphi.^{\rho}(\lambda x : \mathbb{T}^{\tau}_{\tau}\left[\!\left[\tau\right]\!\right].^{\vartheta}\mathbb{T}^{e}_{e}\left[\!\left[e\right]\!\right]_{\vartheta}\ \mathsf{at}\ \rho)\ \mathsf{at}\ \rho \\
\mathbb{T}^{e}_{e}\left[\!\left[e_1\ e_2\right]\!\right]_{\theta} & = & \mathbb{T}^{e}_{e}\left[\!\left[e_1\right]\!\right]\ [\theta]\ \mathbb{T}^{e}_{e}\left[\!\left[e_2\right]\!\right] \\
\mathbb{T}^{e}_{e}\left[\!\left[(e_1, e_2)\ \mathsf{at}\ \rho\right]\!\right]_{\theta} & = & (\mathbb{T}^{e}_{e}\left[\!\left[e_1\right]\!\right]_{\theta}, \mathbb{T}^{e}_{e}\left[\!\left[e_2\right]\!\right]_{\theta})\ \mathsf{at}\ \rho \\
\mathbb{T}^{e}_{e}\left[\!\left[\mathsf{fst}\ e\right]\!\right]_{\theta} & = & \mathsf{fst}\ \mathbb{T}^{e}_{e}\left[\!\left[e\right]\!\right]_{\theta} \\
\mathbb{T}^{e}_{e}\left[\!\left[\mathsf{snd}\ e\right]\!\right]_{\theta} & = & \mathsf{snd}\ \mathbb{T}^{e}_{e}\left[\!\left[e\right]\!\right]_{\theta} \\
\mathbb{T}^{e}_{e}\left[\!\left[\mathsf{letregion}\ \varrho.e\right]\!\right]_{\theta} & = & \mathsf{letregion}\ \varrho.\mathbb{T}^{e}_{e}\left[\!\left[e\right]\!\right]_{\varrho} \\
\mathbb{T}^{e}_{e}\left[\!\left[\lambda\varrho.^{\varphi} u\ \mathsf{at}\ \rho\right]\!\right]_{\theta} & = & \lambda\varrho \succeq \{\}.^{\rho}(\lambda\vartheta \succeq \varphi.^{\vartheta}\mathbb{T}^{e}_{e}\left[\!\left[u\right]\!\right]_{\vartheta}\ \mathsf{at}\ \rho)\ \mathsf{at}\ \rho \\
\mathbb{T}^{e}_{e}\left[\!\left[e\ [\rho]\right]\!\right]_{\theta} & = & \mathbb{T}^{e}_{e}\left[\!\left[e\right]\!\right]_{\theta}\ [\rho]\ [\theta]
\end{array}$$

Programs

$$\begin{array}{rcl}
\mathbb{T}^{p}_{p}\left[\!\left[p\right]\!\right] & = & \mathbb{T}^{e}_{e}\left[\!\left[p\right]\!\right]_{\mathcal{H}}
\end{array}$$

Figure 58: Translation from the Tofte-Talpin Calculus to the Single Effect Calculus

$$\begin{array}{lrcl} \text{Surface terms} & e & ::= & \ldots \mid \mathsf{fix}\ \ f : \tau.u \\ \text{Computation terms} & e & ::= & \ldots \mid \mathsf{fix}\ \ f : \tau.u \end{array}$$

$\boxed{S; e \hookrightarrow S'; e}$

$$\frac{\rho \equiv r \qquad r \in dom(S) \qquad l \notin dom(S(r))}{S; \mathsf{fix}\ f : (\tau_1 \xrightarrow{\theta'} \tau_2, \rho).\lambda x : \tau_1.^{\theta'} e'\ \mathsf{at}\ \rho \hookrightarrow S\{(r,l) \mapsto \lambda x : \tau_1.^{\theta'} e'[\langle l \rangle_r / f]\}; \langle l \rangle_r}$$

$$\frac{\rho \equiv r \qquad r \in dom(S) \qquad l \notin dom(S(r))}{S; \mathsf{fix}\ f : (\Pi \varrho \succeq \varphi.^{\theta'} \tau, \rho).\lambda \varrho \succeq \varphi.^{\theta'} u'\ \mathsf{at}\ \rho \hookrightarrow S\{(r,l) \mapsto \lambda \varrho \succeq \varphi.^{\theta'} u'[\langle l \rangle_r / f]\}; \langle l \rangle_r}$$

$\boxed{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} e : \tau, \theta}$

$$\frac{\Delta; \Gamma, f : \tau; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} u : \tau, \theta}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\text{exp}} \mathsf{fix}\ f : \tau.u : \tau, \theta}$$

Figure 59: Extensions to SEC for fix

(4) If $\Delta \vdash_{\text{btype}} \mu$*, then forall* Δ' *and* ρ*,*
if $\mathbb{T}^{\Delta}_{\Delta} [\![\Delta]\!], \Delta' \vdash_{\text{place}} \rho$*, then* $\mathbb{T}^{\Delta}_{\Delta} [\![\Delta]\!], \Delta' \vdash_{\text{btype}} \mathbb{T}^{\mu}_{\mu} [\![\mu]\!]_{\rho}$.

(5) If $\Delta \vdash_{\text{type}} \tau$*, then forall* Δ'*,*
if $\vdash_{\text{rctxt}} \mathbb{T}^{\Delta}_{\Delta} [\![\Delta]\!], \Delta'$*, then* $\mathbb{T}^{\Delta}_{\Delta} [\![\Delta]\!], \Delta' \vdash_{\text{type}} \mathbb{T}^{\tau}_{\tau} [\![\tau]\!]$.

(6) If $\Delta \vdash_{\text{vctxt}} \Gamma$*, then* $\mathbb{T}^{\Delta}_{\Delta} [\![\Delta]\!] \vdash_{\text{vctxt}} \mathbb{T}^{\Gamma}_{\Gamma} [\![\Gamma]\!]$.

(7) If $\Delta; \Gamma \vdash_{\text{exp}} e : \tau, \varphi$*, then forall* Δ' *and* θ*,*
if $\vdash_{\text{ctxt}} \mathbb{T}^{\Delta}_{\Delta} [\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma} [\![\Gamma]\!]; \theta$ *and* $\mathbb{T}^{\Delta}_{\Delta} [\![\Delta]\!], \Delta' \vdash_{\text{re}} \theta \succeq \varphi$*,*
then $\mathbb{T}^{\Delta}_{\Delta} [\![\Delta]\!], \Delta'; \mathbb{T}^{\Gamma}_{\Gamma} [\![\Gamma]\!] \vdash_{\text{exp}} \mathbb{T}^{e}_{e} [\![e]\!]_{\theta} : \mathbb{T}^{\tau}_{\tau} [\![\tau]\!], \theta$.

(8) If $\vdash_{\text{prog}} p\ \mathsf{ok}$*,*
then $\vdash_{\text{prog}} \mathbb{T}^{p}_{p} [\![p]\!]\ \mathsf{ok}$.

5.3 Region Polymorphic Recursion

Region polymorphic recursion can be supported in the Single Effect Calculus by adding fix and fixing a region abstraction (as is shown by Henglein, Makholm, and Niss [11] for the Tofte-Talpin Calculus); Figure 59 presents the extensions necessary to support fix. The translation given in the previous section is simply extended as follows:

$$\mathbb{T}^{e}_{e} [\![\mathsf{fix}\ f : \tau.u]\!]_{\theta} \quad = \quad \mathsf{fix}\ f : \mathbb{T}^{\tau}_{\tau} [\![\tau]\!] . \mathbb{T}^{e}_{e} [\![u]\!]_{\theta}$$

As an example, consider the following term to compute a factorial (in which we elide the type annotation on *fact*):

```
fix fact.
  (Πϱi ⪰ {}.ρf (Πϱo ⪰ {}.ρf (Πϱb ⪰ {ρf, ϱi, ϱo}.ρf
   (λn : (int, ϱi).ϱb
    if letregion ϱ in n ≤ (1 at ϱ)
     then 1 at ϱo
     else letregion ϱi' in (letregion ϱo' in
          (fact [ϱi'] [ϱo'] [ϱo']
               (letregion ϱ in n − (1 at ϱ) at ϱi'))) ∗ n at ϱo
  ) at ρf) at ρf) at ρf) at ϱf
```

The function *fact* is parameterized by three regions: ϱ_i is the region in which the input integer is allocated, ϱ_o is the region in which the output integer is to be allocated, and ϱ_b is a region that bounds the latent effect of the function. (Region ρ_f is assumed to be bound in an outer context and holds the closure.) We see that the bounds on ϱ_i and ϱ_o indicate that they are not constrained to be outlived by any other regions. On the other hand, the bound on ϱ_b indicates that ρ_f, ϱ_i, and ϱ_o must outlive ϱ_b. Hence, ϱ_b suffices to bound the effects within the body of the function, in which we expect regions ρ_f (at the recursive call) and ϱ_i to be read from and region ϱ_o to be allocated in. Note that the regions passed to the recursive call of *fact* satisfy the bounds, as $\varrho_{o'}$ outlives ρ_f (through $\varrho_{i'}$ and ϱ_b), $\varrho_{i'}$ is allocated before (and deallocated after) $\varrho_{o'}$, and $\varrho_{o'}$ clearly outlives itself.

5.4 Effect Polymorphism

A completely satisfactory account of effect polymorphism in the Single Effect Calculus has proved elusive. Here, we present our thoughts on possible ways of incorporating effect polymorphism into the framework described in this paper.

Recall that effect polymorphism provides a means to abstract over an entire set of regions. Effect instantiation applies an effect abstraction to a set of regions. Effect polymorphism is especially useful for typing higher-order functions. For example, the type of the list `map` function is polymorphic in the effect of the functional argument.

We note that effect polymorphism is most useful in the presence of type polymorphism. While we have presented the Single Effect Calculus (and the Tofte-Talpin region calculus) as monomorphic languages, adding type polymorphism is entirely orthogonal to the development thus far.

Second, we note that effect polymorphism is primarily used to abstract the effect of functions. While the types $\forall\varepsilon.\mathsf{bool}$ or $\forall\varepsilon.((\mathsf{int},\rho_1)\times(\mathsf{int},\rho_2),\rho_3)$ are well-formed, they appear to have utility in a program. On the other hand, the types $\forall\varepsilon.(\tau_1 \xrightarrow{\varphi} \tau_2,\rho)$ and $\forall\varepsilon.(\Pi\varrho.^{\varphi}\tau,\rho)$ are well-formed and have immediate utility in abstracting the effect ε in the latent effect φ or in the latent effects of the types τ_1, τ_2, and τ. It is less clear whether $\forall\varepsilon.(\tau_1\times\tau_2,\rho)$ is of significantly more utility than $(\forall\varepsilon.\tau_1\times\forall\varepsilon.\tau_2,\rho)$.

5.4.1 A Non-compositional Encoding

Effect polymorphism can be simulated in the Single Effect Calculus, although at a heavier notational cost than the encoding of latent effects.

Encoding effect polymorphism in the Single Effect Calculus begins by replacing effect abstractions ($\forall\varepsilon.\tau$) by region abstractions with an empty bound ($\Pi\varepsilon\succeq\{\}.^{\mathcal{H}}\tau$). Effect instantiation (by a set of regions) must be translated to region instantiation; in particular, the set of regions must be translated to a single region denoting the upper bound of the set. In the presence of region polymorphism, this can be complicated, because a set of region variables may have no obvious upper bound. Hence, we must extend the translation to include upper bounds for each set of region variables that may be used in an effect instantiation. For example, a source type like $(\Pi\varrho_1.^{\varphi_1}(\Pi\varrho_2.^{\varphi_2}.(\Pi\varrho_3.^{\varphi_3}\tau,\rho_3),\rho_2),\rho_1)$ (where any subset of $\{\varrho_1,\varrho_2,\varrho_3\}$ may be used in an effect instantiation) is translated to

$$
\begin{array}{l}
(\Pi\varrho_1\succeq\{\}.^{\rho_1}(\Pi\vartheta_1\succeq\varphi_1.^{\rho_1}\\
\ (\Pi\varrho_2\succeq\{\}.^{\rho_2}(\Pi\vartheta_2\succeq\varphi_2.^{\rho_2}\\
\ \ (\Pi\varrho_{12}\succeq\{\varrho_1,\varrho_2\}.^{\rho_2}\\
\ \ \ (\Pi\varrho_3\succeq\{\}.^{\rho_3}(\Pi\vartheta_3\succeq\varphi_3.^{\rho_3}\\
\ \ \ \ (\Pi\varrho_{13}\succeq\{\varrho_1,\varrho_3\}.^{\rho_3}(\Pi\varrho_{23}\succeq\{\varrho_2,\varrho_3\}.^{\rho_3}\\
\ \ \ \ \ (\Pi\varrho_{123}\succeq\{\varrho_1,\varrho_2,\varrho_3\}.^{\rho_3}\\
\ \ \ \ \ \ (\mathbb{T}[\![\tau]\!],\rho_3),\rho_3),\rho_3),\rho_3),\rho_3),\rho_3),\rho_2).\rho_2),\rho_2),\rho_1),\rho_1)
\end{array}
$$

In the term translation, ϱ_{12}, ϱ_{23}, and ϱ_{123} can be used for region instantiations when the source term performs an effect instantiation with the corresponding set of region variables. The burden of instantiating ϱ_{12}, ϱ_{23}, and ϱ_{123} falls to the term that instantiates ϱ_1, ϱ_2, and ϱ_3, which will have sufficient information to choose the right upper bounds.

Unfortunately, this leads to a non-compositional encoding.

5.4.2 A Dynamic Encoding

The difficulty with the above encoding is that every effect instantiation must be translated to a region instantiation, where we must "pick" the region that denotes the upper bound of the set. The real problem is getting this upper bound *region* to the point of the instantiation. This corresponds to the non-compositional aspect of always having the upper-bound of each subset of region variables in scope.

However, there is one key difference between effect abstraction (simulated by region abstraction) and region abstraction. Namely, that it is impossible to *use* any of the regions of an effect variable within the scope of the abstraction.

Suppose we were able to choose this upper bound *dynamically*:

$$\frac{\begin{array}{c}\varphi \equiv \{r_{i_1}, \ldots, r_{i_n}\} \\ S \equiv S_1, r_1 \mapsto R_1, S_2, \ldots, S_n, r_n \mapsto R_n, S_{n+1} \\ S; e[r_n/\varrho] \hookrightarrow S'; v'\end{array}}{S; \mathsf{pick}\ \varrho \succeq \varphi\ \mathsf{in}\ e \hookrightarrow S'; v'} \qquad \frac{\begin{array}{c}\varphi \equiv \{\rho_1, \ldots, \bullet, \ldots, \rho_n\} \\ S; e[\bullet/\varrho] \hookrightarrow S'; v'\end{array}}{S; \mathsf{pick}\ \varrho \succeq \varphi\ \mathsf{in}\ e \hookrightarrow S'; v'}$$

$$\frac{\Delta, \varrho \succeq \varphi; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\exp} e : \tau, \theta}{\Delta; \Gamma; \mathcal{S} : \overline{\mathcal{S}} \vdash_{\exp} \mathsf{pick}\ \varrho \succeq \varphi\ \mathsf{in}\ e : \tau, \theta}$$

The expression $\mathsf{pick}\ \varrho \succeq \varphi\ \mathsf{in}\ e$ instantiates ϱ, at run-time, with the upper bound of the set φ.

Now, one can imagine translating an effect instantiation $e\ [\varphi]$ into $\mathsf{pick}\ \varrho \succeq \varphi\ \mathsf{in}\ e\ [\varrho]$; we have effectively chosen the upper bound needed to capture the effects in φ.

Unfortunately, the dynamic semantics of pick are somewhat unsatisfactory. In particular, within the scope of pick, one can allocate into the chosen region:

$$\mathsf{pick}\ \varrho \succeq \varphi\ \mathsf{in}\ \mathsf{let}\ x = 1\ \mathsf{at}\ \varrho\ \mathsf{in}\ \mathsf{tt}$$

However, if we consider the motivation for introducing pick, we come to a new observation. We introduced pick to choose the right region to instantiate what had been an effect abstraction. So, in an encoding of effect polymorphism, we would not expect to use the chosen region for an allocation (i.e., no $i\ \mathsf{at}\ \varrho$ terms). Therefore, in the dynamic semantics, while we need to pick the upper-bound region to satisfy a type-preserving execution, we do not need that region for a type-erasing execution. This corresponds to the fact that in a type-erasing execution, we need the region handles (which are implicit in the Single Effect Calculus), but not the region types.

Therefore, this suggests that a formulation of the Single Effect Calculus with pick that makes the region / region handle explicit may be able to encode a reasonable facimile of effect polymorphism. The argument for reasonableness is that the pick simply manipulates regions at the type level, without manipulating region handles at the term level (i.e., within the scope of pick, we have the region type ϱ, but not the region handle $\mathsf{handle}(\varrho)$).

5.4.3 An Alternate Translation to $\mathsf{F}^{\mathsf{RGN}}$

An alternate account of effect polymorphism would be to forgoe the translation into the Single Effect Calculus altogether and to adopt a modified formulation of $\mathsf{F}^{\mathsf{RGN}}$. Essentially, we wished to encode effects using a single type for the index of the RGN monad. If we were to instead encode the entire effect as the index of the RGN monad, we may be able to give a cleaner account of effect polymorphism, although this is by no means certain.

One really nice aspect of the current translation is that there are no terms for transformations on RGN computations in the (surface syntax of the) target. The only way to acquire a term of type $\tau_r \preceq \tau_s \equiv$

$\forall\beta.\mathsf{RGN}\ \tau_r\ \beta \to \mathsf{RGN}\ \tau_s\ \beta$ is through letRGN. The two trivial casts (corresponding to reflexivity and transitivity) can be written in System F without any additions:

$$\begin{array}{rcl} \mathit{refl} & :: & \forall r.r \preceq r \\ \mathit{refl} & \equiv & \Lambda r.\Lambda\beta.\lambda k : \mathsf{RGN}\ r\ \beta.k \end{array}$$

$$\begin{array}{rcl} \mathit{trans} & :: & \Lambda r,s,t.(r \preceq s) \to (s \preceq t) \to (r \preceq t) \\ \mathit{trans} & \equiv & \Lambda r,s,t.\lambda f : r \preceq s.\lambda g : s \preceq t.\Lambda\beta.\lambda k : \mathsf{RGN}\ r\ \beta.g\ [\beta]\ (f\ [\beta]\ k) \end{array}$$

If we were to adopt a source calculus where latent effects were given by

$$\varphi ::= \emptyset \mid \{\rho\} \mid \varepsilon \mid \varphi_1 \cup \varphi_2$$

then the "obvious" translation would be something like:

$$\begin{array}{rcl} \mathbb{T}\,[\![\emptyset]\!] & = & \mathsf{unit} \\ \mathbb{T}\,[\![\{\rho\}]\!] & = & \rho \\ \mathbb{T}\,[\![\varepsilon]\!] & = & \varepsilon \\ \mathbb{T}\,[\![\varphi_1 \cup \varphi_2]\!] & = & \mathbb{T}\,[\![\varphi_1]\!] \times \mathbb{T}\,[\![\varphi_2]\!] \end{array}$$

Now we would require some interpretation for the cast judgements like

$$\frac{\Delta \vdash_{\mathrm{ee}} \varphi \supseteq \varphi_1 \qquad \Delta \vdash_{\mathrm{ee}} \varphi \supseteq \varphi_2}{\Delta \vdash_{\mathrm{ee}} \varphi \supseteq \varphi_1 \cup \varphi_2}$$

From a typing perspective this needs to translate to a term with the type

$$\forall e, e_1, e_2.\forall\beta.\mathsf{RGN}\ (e_1 \times e_2)\ \beta \longrightarrow \mathsf{RGN}\ e\ \beta$$

But, certainly there is no System F term with that type. It there fore seems that we would need to introduce a number of witness terms (at the surface syntax) in order to accomodate all of the $\vdash_{\mathrm{er}}$ and $\vdash_{\mathrm{ee}}$ judgements.

Unfortunately, in this system, one can aquire numerous witness functions. And we lose a nice property of the current formulation: namely, that there aren't any non-trivial witness terms – we only aquire one as the by-product of doing something that changes the region on the top of the stack (namely, entering a new region).

6 Related Work

The work in this paper draws heavily from two lines of research. The first is the work done in type-safe region-based memory management, introduced by Tofte and Talpin [25, 26]. Our Single Effect Calculus draws inspiration from the Capability Calculus [5] and Cyclone [8], where the "outlives" relationship between regions is recognized as an important component of type-systems for region calculi.

The work of Banerjee, Heintze and Riecke [2] deserves special mention. They show how to translate the region calculus of Tofte and Talpin into an extension of the polymorphic λ-calculus called $\mathsf{F}_{\#}$. A new type operator # is used as a mechanism to hide and reveal the structure of types. Capabilities to allocate and read values from a region are explicitly passed as polymorphic functions of types $\forall\alpha.\alpha \to (\alpha\#\rho)$ and $\forall\alpha.(\alpha\#\rho) \to \alpha$; however, regions have no run-time significance in $\mathsf{F}_{\#}$ and there is no notion of deallocation upon exiting a region. The equality theory of types in $\mathsf{F}_{\#}$ is nontrivial, due to the treatment of #; in contrast, type equality on $\mathsf{F}^{\mathsf{RGN}}$ types is purely syntactic. Furthermore, their proof of soundness is based on denotational techniques, whereas ours are based on syntactic techniques which tend to scale more easily to other linguistic features. Finally, it is worth noting that there is almost certainly a connection between the $\mathsf{F}_{\#}$ lift and seq expressions and the monadic return and bind operations, although it is not mentioned or explored in the paper.

The second line of research on which we draw is the work done in monadic encapsulation of effects [17, 18, 21, 14, 28, 15, 16, 22, 1, 13, 23, 19, 29]. The majority of this work has focused on effects arising from reading and writing mutable state. While recent work [28, 19, 29] has considered more general combinations of effects and monads, no work has examined the combination of regions and monads.

Launchbury and Peyton Jones [14, 15] introduced a monadic *state transformer* type $\mathsf{ST}\ s\ \alpha$ for computations which transform a state indexed by s and delivers a value of type α. To run such state transforming computations, they provide a term runST with the type $\forall\alpha.(\forall s.\mathsf{ST}\ s\ \alpha) \to \alpha$. Our typing rules for runRGN and letRGN, inspired by that of runST, use the same parametricity to ensure that computations do not leak any (direct or indirect) references to deallocated regions.

Launchbury and Sabry [16] argue that the principle behind runST can be generalized to provide nested scope. They introduce two constants

$$\begin{array}{rcl} \mathsf{blockST} & :: & (\forall\beta.\mathsf{ST}\ (\alpha \times \beta)\ \tau) \to \mathsf{ST}\ \alpha\ \tau \\ \mathsf{importVar} & :: & \mathsf{MutVar}\ \alpha\ \tau \to \mathsf{MutVar}\ (\alpha \times \beta)\ \tau \end{array}$$

where blockST encapsulates a new scope and importVar explicitly allows variables from an enclosing scope to be manipulated by the inner scope. Similarly, Peyton Jones[4] suggests introducing the constant

$$\mathsf{liftST} \quad :: \quad \mathsf{ST}\ \alpha\ \tau \to \mathsf{ST}\ (\alpha \times \beta)\ \tau$$

in lieu of importVar, with the same intention of importing computations from an outer scope into the inner scope. At first glance, this mechanism seems sufficient for supporting a translation from a region calculus. However, in the presence of region polymorphism, such an approach proves difficult. The problem is that the explicit connection between the outer and inner scopes in the product type enforces a total order on regions. This total order is expressed in the types of region allocated values. Hence, one cannot write a function polymorphic in the regions ρ_1 and ρ_2 and apply it in all three of the following situations: (a) instantiate ρ_1 and ρ_2 with the same region, (b) instantiate ρ_1 with a region that strictly outlives the region that instantiates ρ_2, (c) vice versa. To put it another way, the function doesn't know what the region stack is going to look like when it is called – it doesn't know where ρ_1 and ρ_2 are relative to each other or to the top of the stack. Hence, we adopt the approach presented in this paper, where we pass evidence showing that each of the regions is live.

Finally, we note that Wadler and Thiemann [29] advocate marrying effects and monads by translating a type $\tau_1 \xrightarrow{\sigma} \tau_2$ to the type $\mathbb{T}\ [\![\tau_1]\!] \to \mathsf{T}^{\sigma}\ \mathbb{T}\ [\![\tau_2]\!]$, where $\mathsf{T}^{\sigma}\ \tau$ represents a computation that yields a value of type τ and has effects delimited by (the set) σ. As with the work of Banerjee et. al. described above,

[4] private communication

this introduces a nontrivial theory of equality (and subtyping) on types; the types $\mathsf{T}^{\sigma}\ \tau$ and $\mathsf{T}^{\sigma'}\ \tau$ are equal so long as σ and σ' are (encodings of) equivalent sets. However, few programming languages allow one to express such nontrivial equalities between types.

7 Conclusions and Future Work

We have given a type- and meaning-preserving translation from the Single Effect Calculus to $\mathsf{F}^{\mathsf{RGN}}$. Both the source and the target calculi use a static type-system to delimit the effects of allocating in and reading from regions. The Single Effect Calculus uses the partial order implied by the "outlives" relation on regions to use single regions as bounds for sets of effects. We feel that this is an important insight that leads to a relatively straight-forward translation into the monadic setting. $\mathsf{F}^{\mathsf{RGN}}$ draws from the work on monadic encapsulation of state to give parametric types to runRGN and letRGN that prevent access of regions beyond their lifetimes. Explicit functions witness the outlives relationship between regions, enabling computations from outer regions to be cast to computations in inner regions. Witness functions cannot be forged and are only introduced via letRGN.

There are numerous directions for future work. One idea is to provide the RGN monad to Haskell programmers and to try to leverage type classes so that witnesses and handles can be passed implicitly, thereby reducing the notational overhead of programming with nested stores. Unfortunately, a direct encoding of the subtyping leads to overlapping instances for witness generation, and we have been unable to find a suitable set of type-class definitions that works around this problem. Obviously, a language that incorporates subtyping directly, such as $\mathsf{F}_{\leq}$, would simplify the encoding.

Finally, as is well known, Tofte and Talpin's original region calculus can lead to inefficient memory usage for some programs. In practice, additional mechanisms are required to achieve good space utilization. Cyclone incorporates a number of these enhancements, including unique pointers and dynamic regions, and it remains to be seen whether these features can also be encoded into a simpler setting.

References

[1] Z. Ariola and A. Sabry. Correctness of monadic state: An imperative call-by-need calculus. In *Proceedings of the 25th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'98)*, pages 62–74, San Diego, CA, 1998. ACM Press.

[2] A. Banerjee, N. Heintze, and J. Riecke. Region analysis and the polymorphic lambda calculus. In *Proceedings of the 14th IEEE Symposium on Logic in Computer Science (LCS'99)*, pages 88–97, Trento, Italy, 1999. IEEE Computer Society Press.

[3] C. Calcagno. Stratified operational semantics for safety and correctness of the region calculus. In *Proceedings of the 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'01)*, pages 155–165, London, England, 2001. ACM Press.

[4] C. Calcagno, S. Helsen, and P. Thiemann. Syntactic type soundness results for the region calculus. *Information and Computation*, 173(2):199–332, Mar. 2002.

[5] K. Crary, D. Walker, and G. Morrisett. Typed memory management in a calculus of capabilities. In *Proceedings of the 26th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'99)*, pages 262–275. ACM Press, 1999.

[6] M. Elsman. Garbage collection safety for region-based memory management. In *Proceedings of the ACM SIGPLAN Workshop on Types in Language Design and Implementation (TLDI'03)*, pages 123–134, New Orleans, LA, 2003. ACM Press.

[7] J.-Y. Girard, P. Taylor, and Y. Lafont. *Proofs and Types*. Cambridge University Press, 1989.

[8] D. Grossman, G. Morrisett, Y. Wang, T. Jim, M. Hicks, and J. Cheney. Formal type soundness for Cyclone's region system. Technical Report 2001-1856, Department of Computer Science, Cornell University, Nov. 2001.

[9] N. Hallenberg, M. Elsman, and M. Tofte. Combining region inference and garbage collection. In *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'02)*, pages 141–152. ACM Press, 2002. Berlin, Germany.

[10] S. Helsen and P. Thiemann. Syntactic type soundness for the region calculus. In *Proceedings of the 4th International Workshop on Higher Order Operational Techniques in Semantics (HOOTS'00)*, volume 41 of *Electronic Notes in Theoretical Computer Science*, pages 1–19, Montreal, Canada, Sept. 2000. Elsevier Science Publishers.

[11] F. Henglein, H. Makholm, and H. Niss. Effect types and region-based memory management. In B. Pierce, editor, *Advanced Topics in Types and Programming Languages*, chapter 5. MIT Press, 2003. In preparation.

[12] M. Hicks, G. Morrisett, D. Grossman, and T. Jim. Safe and flexible memory management in Cyclone. Technical Report CS-TR-4514, University of Maryland Department of Computer Science, July 2003.

[13] R. Kieburtz. Taming effects with monadic typing. In *Proceedings of the 3rd ACM SIGPLAN International Conference on Functional Programming (ICFP'98)*, pages 51–62, Baltimore, MD, 1998. ACM Press.

[14] J. Launchbury and S. Peyton Jones. State in Haskell. *Lisp and Symbolic Computation*, 8(4):293–341, 1995.

[15] J. Launchbury and S. Peyton Jones. Lazy functional state threads. In *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'94)*, pages 24–35, Orlando, FL, 1997. ACM Press.

[16] J. Launchbury and A. Sabry. Monadic state: Axiomatization and type safety. In *Proceedings of the 2nd ACM SIGPLAN International Conference on Functional Programming (ICFP'97)*, pages 227–237, Amsterdam, The Netherlands, 1997. ACM Press.

[17] E. Moggi. Computational lambda calculus and monads. In *Proceedings of the 4th IEEE Symposium on Logic in Computer Science (LCS'89)*, pages 14–23, Pacific Grove, CA, 1989.

[18] E. Moggi. Notions of computation and monads. *Information and Computation*, 93(1):55–92, Jan. 1991.

[19] E. Moggi and A. Sabry. Monadic encapsulation of effects: a revised approach (extended version). *Journal of Functional Programming*, 11(6):591–627, Nov. 2001.

[20] J. Reynolds. Towards a theory of type structure. In *Programming Symposium*, volume 19 of *Lecture Notes in Computer Science*, pages 408–425, Paris, France, Apr. 1974. Springer-Verlag.

[21] J. Riecke and R. Viswanathan. Isolating side effects in sequential languages. In *Proceedings of the 22nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'95)*, pages 1–12, San Francisco, CA, 1995. ACM Press.

[22] A. Sabry and P. Wadler. A reflection on call-by-value. *ACM Transactions on Programming Languages and Systems*, 19(6):916–941, Nov. 1997.

[23] M. Semmelroth and A. Sabry. Monadic encapsulation in ml. In *Proceedings of the 4th ACM SIGPLAN International Conference on Functional Programming (ICFP'99)*, pages 8–17, Paris, France, 1999. ACM Press.

[24] G. Smith and D. Volpano. A sound polymorphic type system for a dialect of c. *Science of Computer Programming*, 32(1-3):49–72, 1998.

[25] M. Tofte and J.-P. Talpin. Implementation of the typed call-by-value λ-calculus using a stack of regions. In *Proceedings of the 21st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'94)*, pages 188–201, Portland, OR, 1994. ACM Press.

[26] M. Tofte and J.-P. Talpin. Region-based memory management. *Information and Computation*, 132(2):109–176, Feb. 1997.

[27] D. Volpano and G. Smith. Eliminating covert flows with minimum typings. In *Proceedings of the 10th IEEE Computer Security Foundations Workshop (CFSW'97)*, pages 156–168. IEEE Computer Society Press, 1997. Rockport, MA.

[28] P. Wadler. The marriage of effects and monads. In *Proceedings of the 3rd ACM SIGPLAN International Conference on Functional Programming (ICFP'98)*, pages 63–74, Baltimore, MD, 1995. ACM Press.

[29] P. Wadler and P. Thiemann. The marriage of effects and monads. *Transactions on Computational Logic*, 4(1):1–32, 2003.